

**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ**  
**УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ НАУКИ І ТЕХНОЛОГІЙ**

**Кафедра «Технологія машинобудування»**

**ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ**

**Методичні вказівки до виконання лабораторного практикуму**  
**для студентів спеціальності**  
**131 – Прикладна механіка (бакалаврський рівень)**

**Частина I**

**Дніпро 2023**

УДК 621.9.06

Об'єктно-орієнтоване програмування : методичні вказівки до виконання лабораторного практикуму для студентів спеціальності 131 – Прикладна механіка (бакалаврський рівень). Частина I / Уклад. : Гришин В. С., Карабут В. М. – Дніпро : Україн. держ. ун-т науки і технол., 2023. – 118 с.

Наведені методика і послідовність проведення лабораторних робіт, вимоги до оформлення звіту з лабораторних робіт, запитання з самоконтролю до розділів, рекомендована література.

Призначені для студентів спеціальності 131 – Прикладна механіка (бакалаврський рівень).

Методичні вказівки ухвалено Групою забезпечення якості освітньої програми «Технологія машинобудування» спеціальності 131 – Прикладна механіка (бакалаврський рівень) Протокол №7 від 14.03.2023 р.

Укладачі:	В. С. Гришин, канд. техн. наук, доц. В. М. Карабут, ст. викладач
Відповідальний за випуск	В. С. Гришин, канд. техн. наук, доц.
Рецензент	Г. І. Толстіков, канд. техн. наук, доц.
Гарант освітньої програми	В. С. Гришин, канд. техн. наук, доц.

Підписано по друку 02.05.2023. Формат 60×84 1/16. Папір друк. Друк плоский. Облік.-вид. арк. 2,83. Умов. друк. арк. 6,86. Замовлення № 31.

Український державний університет науки і технологій  
49010, м. Дніпро, вул. Лазаряна, 2  
Редакційно-видавничий відділ УДУНТ

© Гришин В. С., Карабут В. М.,  
укладання, 2023

© Україн. держ. ун-т науки і технол.,  
оригінал-макет, 2023

## ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. ОСНОВИ МОВИ ПРОГРАМУВАННЯ DELPHI.....	5
Лабораторна робота №1. Тема: «Блок-схема алгоритму з лінійною структурою».....	5
Лабораторна робота №2. Тема: «Блок-схема алгоритму з розгалужуючою структурою».....	12
Лабораторна робота №3. Тема: «Лінійна структура програми і типи даних».....	21
Лабораторна робота №4. Тема: «Розгалужуюча структура програми і типи даних».....	36
РОЗДІЛ 2. МЕТОДИ ВВОДУ ДАНИХ.....	50
Лабораторна робота №5. Тема: «Математичні функції і процедури».....	50
Лабораторна робота №6. Тема: «Діалоговий ввід-вивід даних».....	66
РОЗДІЛ 3. МЕТОДИ ВИБОРУ ДАНИХ.....	78
Лабораторна робота №7. Тема: «Структури прийняття рішень. Оператор if».....	78
Лабораторна робота №8. Тема: «Структури прийняття рішень. Оператор case».....	91
Література.....	102
Додаток А Форма звіту з лабораторної роботи №1.....	103
Додаток Б Форма звіту з лабораторної роботи №2.....	105
Додаток В Форма звіту з лабораторної роботи №3.....	107
Додаток Г Форма звіту з лабораторної роботи №4.....	109
Додаток Д Форма звіту з лабораторної роботи №5.....	111
Додаток Є Форма звіту з лабораторної роботи №6.....	113
Додаток Ж Форма звіту з лабораторної роботи №7.....	115
Додаток З Форма звіту з лабораторної роботи №8.....	117

## ВСТУП

Навчальна дисципліна «Об'єктно-орієнтоване програмування» є вибірковою для вивчення студентами, які здобувають освітній ступінь бакалавра за освітньою програмою «Технологія машинобудування».

Мета навчальної дисципліни – засвоєння початкових знань та придбання навичок, необхідних для розробки програмних продуктів комп'ютерного забезпечення в технологічних процесах обробки металів різанням.

Очікувані результати навчання за навчальною дисципліною наступні:

- планувати розробку програм. Вибирати типи даних. Знаходити змінні і константи. Застосовувати приведення типів даних;
- розробляти і використовувати програмне забезпечення та методи вводу даних в технологічних процесах обробки металів різанням;
- розробляти і використовувати програмне забезпечення та методи вибору даних в технологічних процесах обробки металів різанням;
- розробляти і використовувати програмне забезпечення та методи обробки даних в технологічних процесах обробки металів різанням;
- розробляти і використовувати програмне забезпечення та види структурних даних в технологічних процесах обробки металів різанням;
- розробляти і використовувати програмне забезпечення та методи виводу даних в технологічних процесах обробки металів різанням;
- розробляти і використовувати програмне забезпечення та компоненти і властивості графічних моделей в технологічних процесах обробки металів різанням.

Передумовами вивчення дисципліни є базові знання з дисциплін: «Вища математика», «Комп'ютерні технології та програмування».

# РОЗДІЛ 1. ОСНОВИ МОВИ ПРОГРАМУВАННЯ DELPHI

## Лабораторна робота №1

**Тема:** Блок-схема алгоритму з лінійною структурою.

**Мета роботи:** Засвоєння початкових знань та придбання навичок, необхідних для розробки програмних продуктів комп'ютерного забезпечення в технологічних процесах обробки металів різанням.

### Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Виконати алгоритм виконання лабораторної роботи №1.
3. Скласти звіт з лабораторної роботи №1 за наведеною формою (див. Додаток А).

#### Хід виконання роботи

##### 1. Теоретичні відомості

Вивчити теоретичні відомості зі списку наведеної літератури.

##### 2. Алгоритм виконання лабораторної роботи №1

###### 2.1. Запуск програми Microsoft Word

У операційній системі Windows, у системному меню «*Пуск*», вибрати:  
**Пуск > Програми > Microsoft Office > Microsoft Word**

###### 2.2. Збереження документа

Після запуску програми Microsoft Word створюється новий документ. Збережемо його. Для цього на панелі «*Стандартна*» натисніть кнопку «*Зберегти*». У з'явившомуся діалоговому вікні «*Збереження документа*» створіть нову папку та присвойте їй ім'я **Лаб\_роб\_№1**. Потім відкрийте цю папку. Далі введіть ім'я файлу **Блок-схема алгоритму з лінійною структурою** і натисніть кнопку «*Зберегти*».

### 2.3. Складання блок-схеми алгоритму з лінійною структурою

Скласти блок-схему алгоритму обчислення вартості поїздки на автомобілі на дачу (туди і назад). Вихідними даними є: відстань до дачі (туди і назад) у км; кількість бензину, яку споживає автомобіль на 100 км пробігу у літрах; ціна одного літра бензину у гривнях.

Приймаємо позначення:  $rast$  – відстань до дачі (туди і назад) у км;  $potr$  – кількість бензину, яку споживає автомобіль на 100 км пробігу у літрах;  $cena$  – ціна одного літра бензину у гривнях.

Складаємо формулу:

$$sum = 2 \cdot (potr/100) \cdot rast \cdot cena$$

Блок-схема алгоритму обчислення вартості поїздки на автомобілі на дачу (туди і назад) наведено на рисунку 2.1.

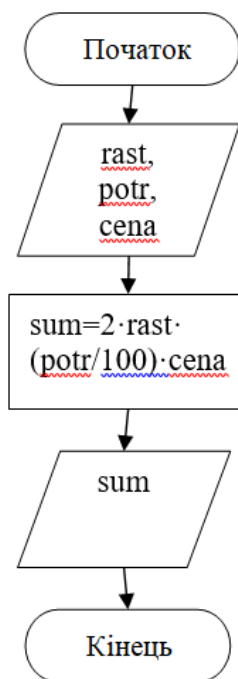


Рис. 2.1. Блок-схема алгоритму обчислення вартості поїздки на автомобілі на дачу (туди і назад)

#### 2.3.1. Побудова полотна для створення блок-схеми

Необхідно виконати такі дії:

- у «**Головному**» меню вибрати вкладку «**Вставка**» (рис. 2.2);
- на вкладці «**Вставка**» натиснути кнопку **Фігури** (рис. 2.2);

в) у з'явившомуся списку натиснути кнопку **Нове полотно** (рис. 2.2);  
г) вказати курсором миші місце розташування полотна. Полотно буде побудовано (рис. 2.3).

**Примітка:**

- використовуючи маркери на полотні можна змінити розмір полотна;
- використовуючи виділення полотна можна перемістити полотно.

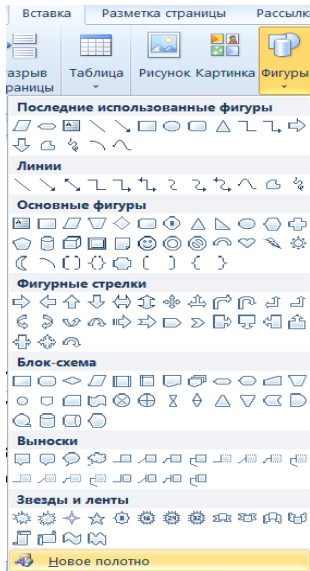


Рис. 2.2. Натиснути кнопку **Нове полотно**

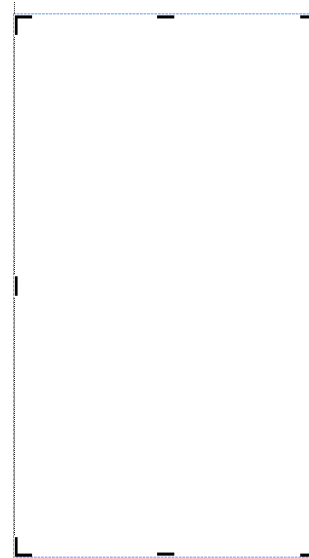


Рис. 2.3. Побудоване полотно

### 2.3.2. Побудова символу блок-схеми початок та закінчення програми

Необхідно виконати такі дії:

- на вкладці «**Вставка**» натиснути кнопку **Фігури** (рис. 2.2);
- у з'явившомуся списку в розділі «**Блок-схема**» натиснути кнопку **Блок-схема: знак завершення** (рис. 2.4);
- вказати курсором миші лівий верхній кут розташування символу, а потім, не відпускаючи ліву кнопку миші, перемістити мишу і вказати правий нижній кут символу. Символ буде побудовано (рис. 2.5);
- для зняття виділення символу необхідно вказати курсором миші у будь-якому місці полотна.

**Примітка:**

- використовуючи маркери символу можна змінити розмір символу;
- використовуючи виділення символу можна перемістити символ.

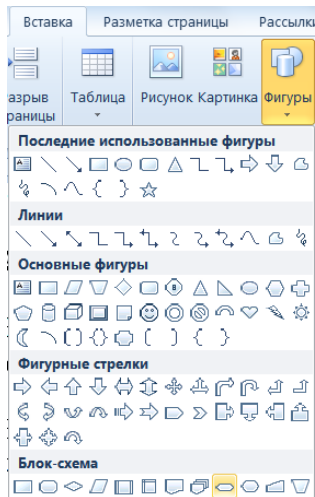


Рис. 2.4. Натиснути кнопку  
Блок-схема: знак завершення

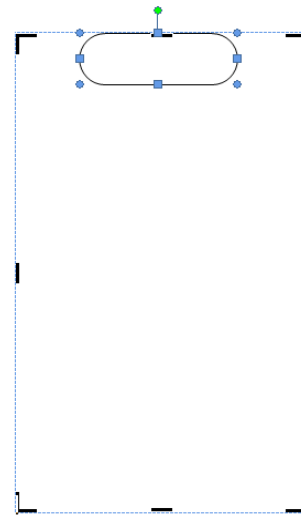


Рис. 2.5. Побудований символ  
Початок та закінчення програми

### 2.3.3. Побудова символу блок-схеми вводу та виводу даних у програмі

Необхідно виконати такі дії:

- а) на вкладці «**Вставка**» натиснути кнопку **Фігури** (рис. 2.2);
- б) у з'явившомуся списку в розділі «**Блок-схема**» натиснути кнопку **Блок-схема: дані** (рис. 2.6);
- в) вказати курсором миші лівий верхній кут розташування символу, а потім, не відпускаючи ліву кнопку миші, перемістити мишу і вказати правий нижній кут символу. Символ буде побудовано (рис. 2.7);
- г) для зняття виділення символу необхідно вказати курсором миші у будь-якому місці полотна.

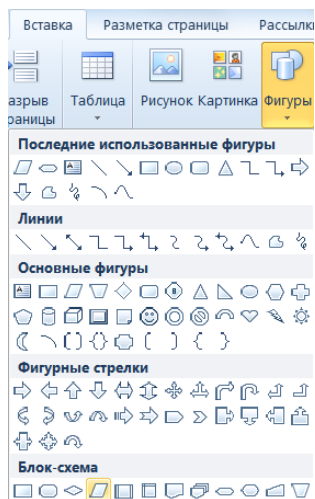


Рис. 2.6. Натиснути кнопку  
Блок-схема: дані

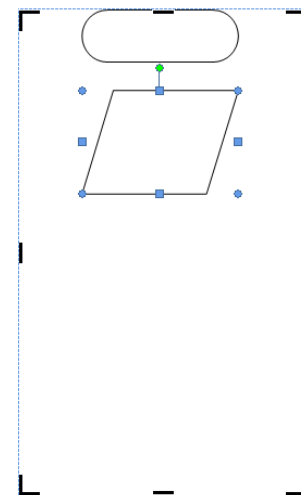


Рис. 2.7. Побудований символ  
вводу та виводу даних

### 2.3.4. Побудова символу блок-схеми обробки даних у програмі

Необхідно виконати такі дії:

а) на вкладці «**Вставка**» натиснути кнопку **Фігури** (рис. 2.2);

б) у з'явившомуся списку в розділі «**Блок-схема**» натиснути кнопку **Блок-схема: процес** (рис. 2.8);

в) вказати курсором миші лівий верхній кут розташування символу, а потім, не відпускаючи ліву кнопку миші, перемістити мишу і вказати правий нижній кут символу. Символ буде побудовано (рис. 2.9);

г) для зняття виділення символу необхідно вказати курсором миші у будь-якому місці полотна.

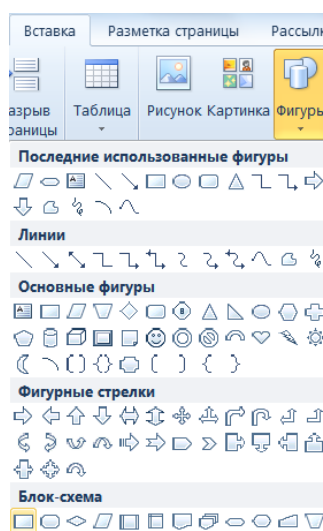


Рис. 2.8. Натиснути кнопку **Блок-схема: процес**

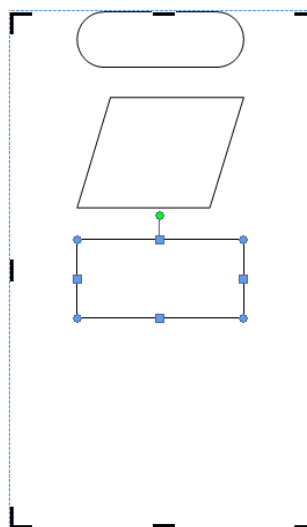


Рис. 2.9. Побудований символ обробки даних

### 2.3.5. Побудова символу блок-схеми вводу та виводу даних у програму

Аналогічно побудувати символ блок-схеми вводу та виводу даних у програму (рис. 2.10), як у п. п. 2.3.3.

### 2.3.6. Побудова символу блок-схеми початок та закінчення програми

Аналогічно побудувати символ блок-схеми початок та закінчення програми (рис. 2.11), як у п. п. 2.3.2.

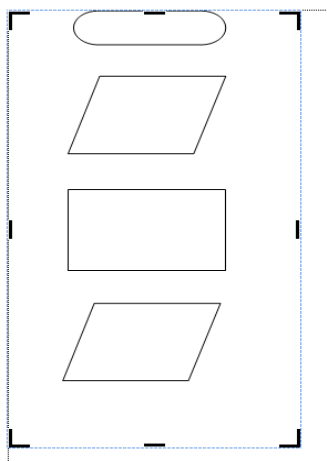


Рис. 2.10. Побудований символ вводу та виводу даних

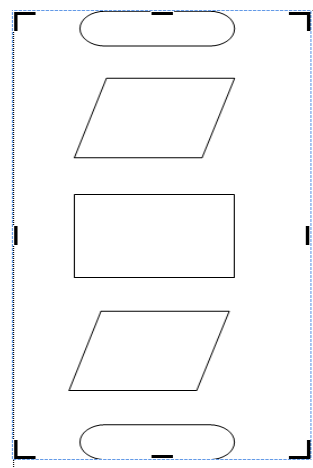


Рис. 2.11. Побудований символ початок та закінчення

### 2.3.7. Побудова символу блок-схеми стрілка програми

Необхідно виконати такі дії:

- а) на вкладці «**Вставка**» натиснути кнопку **Фігури** (рис. 2.2);
- б) у з'явившомуся списку в розділі «**Линії**» натиснути кнопку **Стрілка** (рис. 2.12);
- в) вказати курсором миші точку на середині нижнього елемента **символу блок-схеми початок і закінчення** – це початкова точка стрілки, а потім вказати курсором миші точку на середині верхнього елемента **символу блок-схеми вводу та виводу даних** – це кінцева точка стрілки (рис. 2.13).

Аналогічно побудувати **символ блок-схеми стрілка** програми до інших символів блок-схеми (рис. 2.14).

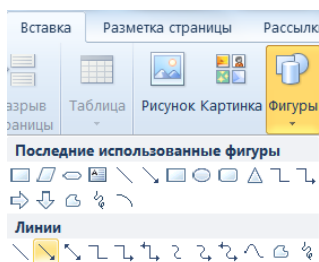


Рис. 2.12. Натиснути кнопку **Стрілка**

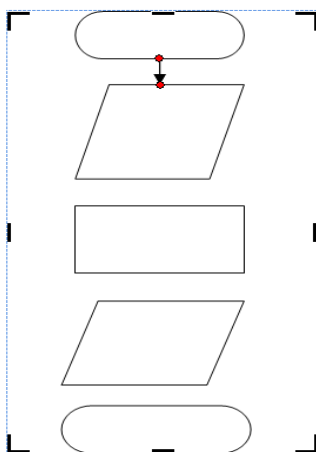


Рис. 2.13. Побудований символ **стрілка**

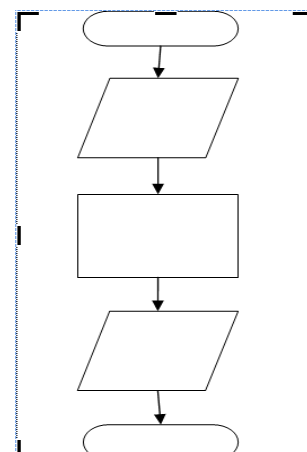


Рис. 2.14. Побудовані символи **стрілка** до інших символів блок-схеми

### 2.3.8. Заповнення текстом символів блок-схеми

Необхідно виконати такі дії:

а) виділити символ **початок та закінчення** програми, вказавши курсором миші на символі **початок та закінчення** програми (рис. 2.15);

б) натиснути праву кнопку миші на символі **початок та закінчення** програми. З'явиться контекстне меню (рис. 2.15);

в) із контекстного меню вибрати **Додати текст** (рис. 2.15);

г) у символі **початок та закінчення** з'явиться поле для вводу тексту. Ввести текст **Початок** (рис. 2.16);

д) для виходу з поля для вводу тексту в символі **початок та закінчення** необхідно вказати курсором миші у будь-якому місці документа.

Аналогічно додати текст до інших символів блок-схеми (рис. 2.17).

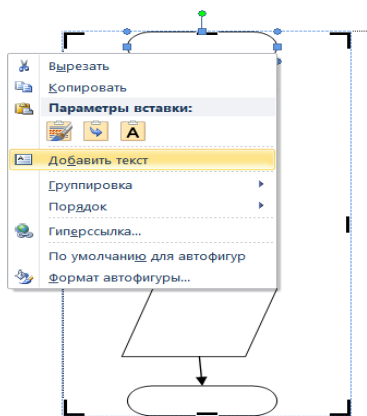


Рис. 2.15. Виділення символу **початок та закінчення** та вибір **Додати текст**

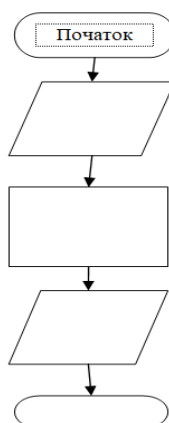


Рис. 2.16. Ввід тексту **Початок**

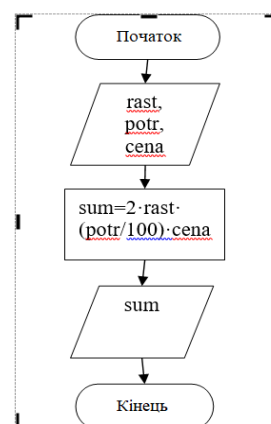


Рис. 2.17. Додавання тексту до інших символів блок-схеми

## 3. Зміст звіту з лабораторної роботи №1

**Звіт** з лабораторної роботи №1 складається з:

- титульного аркуша (див. Додаток А);
- аркуша (див. Додаток А) на якому розташовано: **тема** лабораторної роботи, **мета** лабораторної роботи, **складання блок-схеми** алгоритму з лінійною структурою (навести **умову** та **формулу**, які розглядаються в лабораторній роботі №1, див. п. п. 2.3).

- аркуша (див. Додаток А) на якому розташовано: **блок-схема** алгоритму з лінійною структурою (навести **блок-схему**, яка розглядається в лабораторній роботі №1, див. п. п. 2.3) та **висновки**.

## Лабораторна робота №2

**Тема:** Блок-схема алгоритму з розгалужуючою структурою.

**Мета роботи:** Засвоєння початкових знань та придбання навичок, необхідних для розробки програмних продуктів комп'ютерного забезпечення в технологічних процесах обробки металів різанням.

### Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Виконати алгоритм виконання лабораторної роботи №2.
3. Скласти звіт з лабораторної роботи №2 за наведеною формою (див. Додаток Б).

### Хід виконання роботи

#### 1. Теоретичні відомості

Вивчити теоретичні відомості зі списку наведеної літератури.

#### 2. Алгоритм виконання лабораторної роботи №2

##### 2.1. Запуск програми Microsoft Word

У операційній системі Windows, у системному меню «*Пуск*», вибрати:  
**Пуск > Програми > Microsoft Office > Microsoft Word**

##### 2.2. Збереження документа

Після запуску програми Microsoft Word створюється новий документ. Збережемо його. Для цього на панелі «*Стандартна*» натисніть кнопку «*Зберегти*». У з'явившомуся діалоговому вікні «*Збереження документа*» створіть нову папку та присвойте їй ім'я **Лаб\_раб\_№2**. Потім відкрийте цю папку. Далі введіть ім'я файлу **Блок-схема алгоритму з розгалужуючою структурою** і натисніть кнопку «*Зберегти*».

## 2.3. Складання блок-схеми алгоритму з розгалужуючою структурою

Скласти блок-схему алгоритму обчислення частки двох чисел. Програма повинна перевіряти правильність введених користувачем даних і якщо вони невірні (ділитель дорівнює нулю), видавати повідомлення про помилку.

Приймаємо позначення:  $a$  – ділене;  $b$  – дільник;  $c$  – приватне.

Складаємо формулу:

$$a/b=c$$

Блок-схема алгоритму обчислення частки двох чисел наведено на рисунку 2.1.

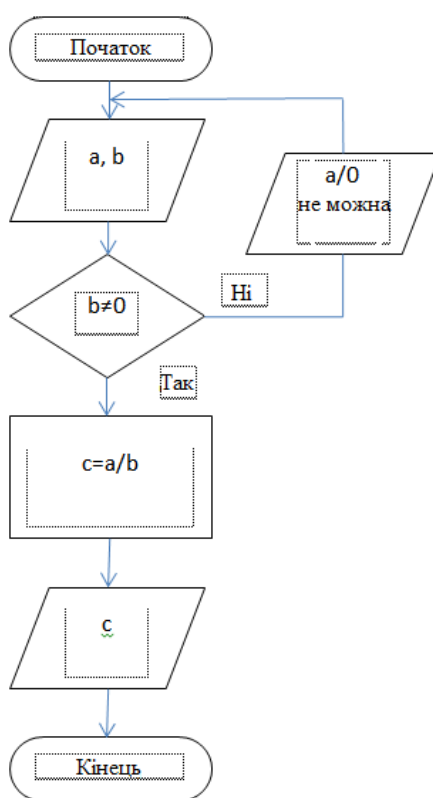


Рис. 2.1. Блок-схема алгоритму обчислення частки двох чисел

### 2.3.1. Побудова полотна для створення блок-схеми

Необхідно виконати такі дії:

- у «Головному» меню вибрати вкладку «Вставка» (рис. 2.2);
- на вкладці «Вставка» натиснути кнопку **Фігури** (рис. 2.2);
- у з'явившомуся списку натиснути кнопку **Нове полотно** (рис. 2.2);
- вказати курсором миші місце розташування полотна. Полотно буде побудовано (рис. 2.3).

### Примітка:

- використовуючи маркери на полотні можна змінити розмір полотна;
- використовуючи виділення полотна можна перемістити полотно.

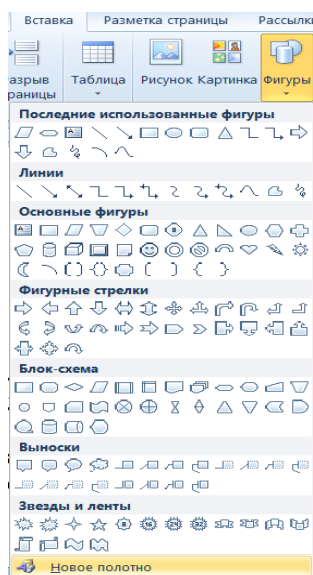


Рис. 2.2. Натиснути кнопку **Нове полотно**



Рис. 2.3. Побудоване полотно

### 2.3.2. Побудова символу блок-схеми початок та закінчення програми

Необхідно виконати такі дії:

- на вкладці «**Вставка**» натиснути кнопку **Фігури** (рис. 2.2);
- у з'явившомуся списку в розділі «**Блок-схема**» натиснути кнопку **Блок-схема: знак завершення** (рис. 2.4);
- вказати курсором миші лівий верхній кут розташування символу, а потім, не відпускаючи ліву кнопку миші, перемістити мишу і вказати правий нижній кут символу. Символ буде побудовано (рис. 2.5);
- для зняття виділення символу необхідно вказати курсором миші у будь-якому місці полотна.

### Примітка:

- використовуючи маркери символу можна змінити розмір символу;
- використовуючи виділення символу можна перемістити символ.

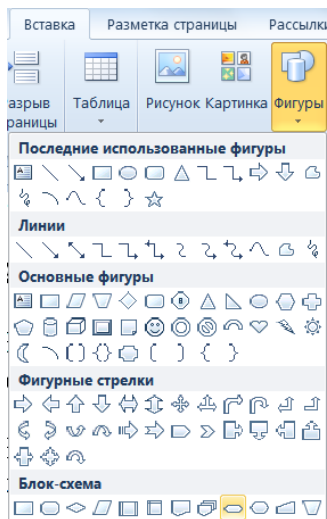


Рис. 2.4. Натиснути кнопку **Блок-схема: знак завершення**

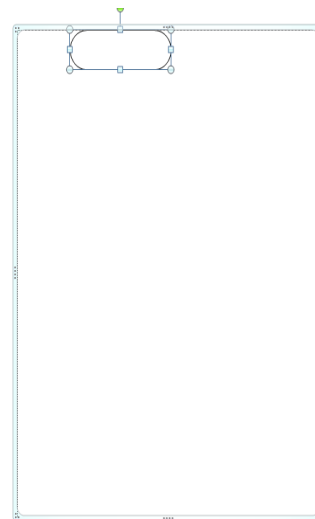


Рис. 2.5. Побудований символ початок та закінчення програми

### 2.3.3. Побудова символу блок-схеми вводу та виводу даних у програмі

Необхідно виконати такі дії:

- а) на вкладці «**Вставка**» натиснути кнопку **Фігури** (рис. 2.2);
- б) у з'явившомуся списку в розділі «**Блок-схема**» натиснути кнопку **Блок-схема: дані** (рис. 2.6);
- в) вказати курсором миші лівий верхній кут розташування символу, а потім, не відпускаючи ліву кнопку миші, перемістити мишу і вказати правий нижній кут символу. Символ буде побудовано (рис. 2.7);
- г) для зняття виділення символу необхідно вказати курсором миші у будь-якому місці полотна.

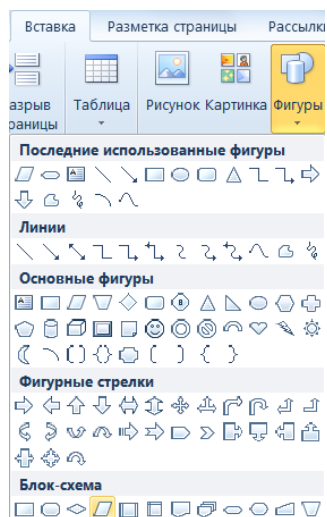


Рис. 2.6. Натиснути кнопку **Блок-схема: дані**

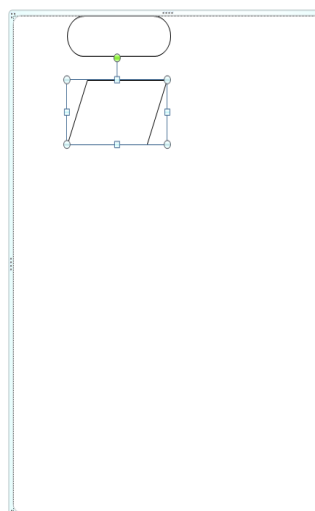


Рис. 2.7. Побудований символ вводу та виводу даних

### 2.3.4. Побудова символу блок-схеми прийняття рішень у програмі

Необхідно виконати такі дії:

а) на вкладці «**Вставка**» натиснути кнопку **Фігури** (рис. 2.2);

б) у з'явившомуся списку в розділі «**Блок-схема**» натиснути кнопку **Блок-схема: рішення** (рис. 2.8);

в) вказати курсором миші лівий верхній кут розташування символу, а потім, не відпускаючи ліву кнопку миші, перемістити мишу і вказати правий нижній кут символу. Символ буде побудовано (рис. 2.9);

г) для зняття виділення символу необхідно вказати курсором миші у будь-якому місці полотна.

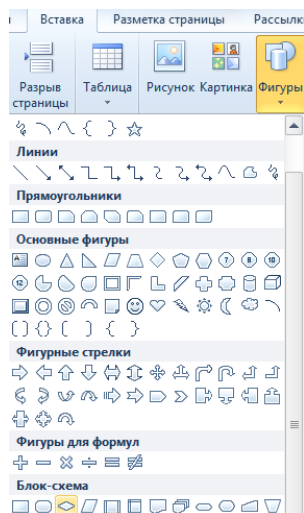


Рис. 2.8. Натиснути кнопку **Блок-схема: рішення**

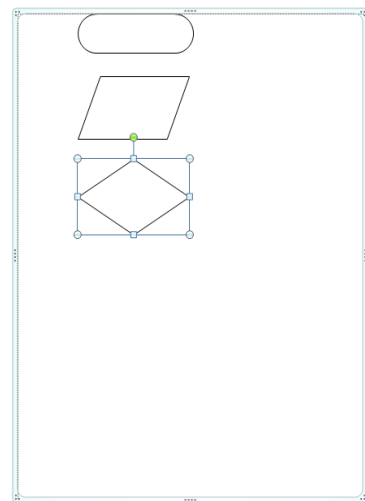


Рис. 2.9. Побудований символ прийняття рішення

### 2.3.5. Побудова символу блок-схеми обробки даних у програмі

Необхідно виконати такі дії:

а) на вкладці «**Вставка**» натиснути кнопку **Фігури** (рис. 2.2);

б) у з'явившомуся списку в розділі «**Блок-схема**» натиснути кнопку **Блок-схема: процес** (рис. 2.10);

в) вказати курсором миші лівий верхній кут розташування символу, а потім, не відпускаючи ліву кнопку миші, перемістити мишу і вказати правий нижній кут символу. Символ буде побудовано (рис. 2.11).

г) для зняття виділення символу необхідно вказати курсором миші у будь-якому місці полотна.

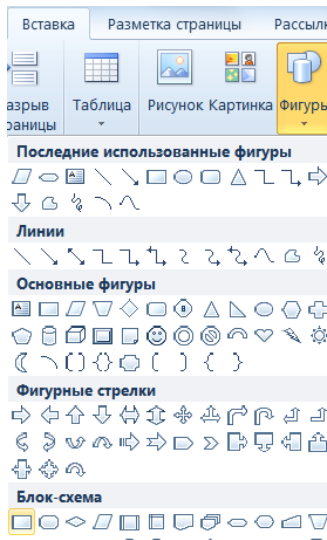


Рис. 2.10. Натиснути кнопку **Блок-схема: процес**

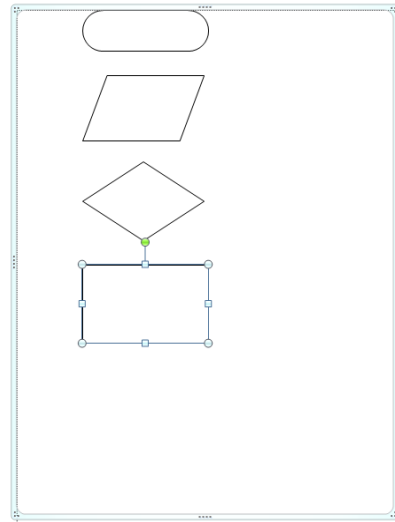


Рис. 2.11. Побудований символ **обробки даних**

### 2.3.6. Побудова символів блок-схеми вводу та виводу даних у програму

Аналогічно побудувати **символи блок-схеми вводу та виводу даних у програму** (рис. 2.12), як у п. п. 2.3.3.

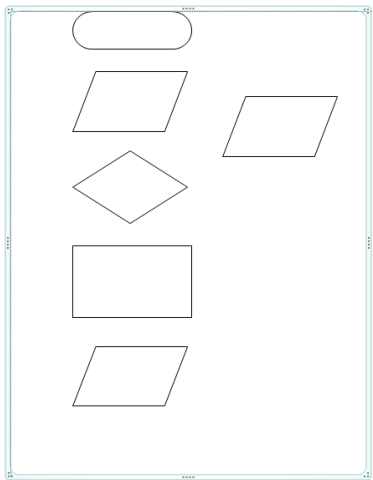


Рис. 2.12. Побудований символ **вводу та виводу даних**

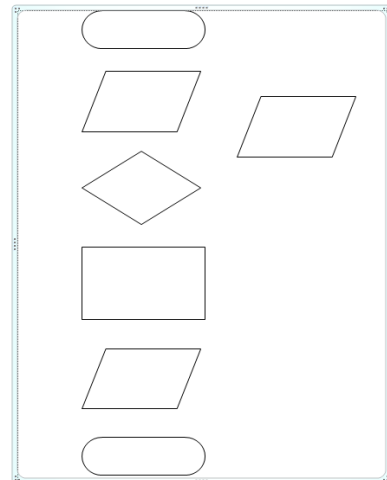


Рис. 2.13. Побудований символ **початок та закінчення**

### 2.3.7. Побудова символу блок-схеми початок та закінчення програми

Аналогічно побудувати символ блок-схеми **початок та закінчення програми** (рис. 2.13), як у п. п. 2.3.2.

### 2.3.8. Побудова символу блок-схеми лінія програми

Необхідно виконати такі дії:

а) на вкладці «**Вставка**» натиснути кнопку **Фігури** (рис. 2.2);

б) у з'явившомуся списку в розділі «**Лінії**» натиснути кнопку **Лінія** (рис. 2.14);

в) вказати курсором миші точку на куті **символу блок-схеми прийняття рішення** – це початкова точка лінії, а потім вказати курсором миші точку на середині нижнього елемента **символу блок-схеми вводу та виводу даних** – це кінцева точка лінії (рис. 2.15).

Аналогічно побудувати **символ блок-схеми лінія програми** до інших символів блок-схеми (рис. 2.16).

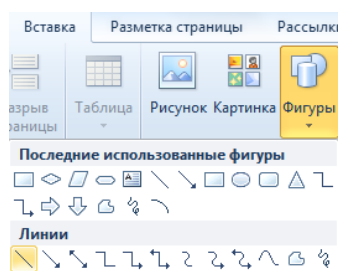


Рис. 2.14. Натиснути кнопку **Лінія**

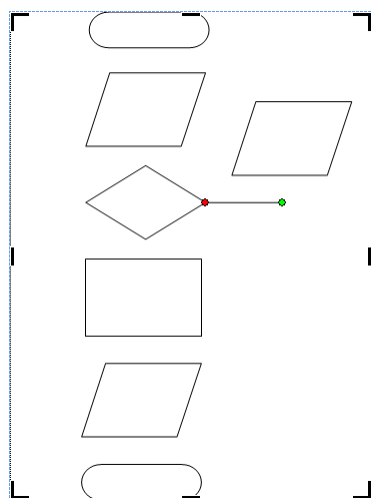


Рис. 2.15. Побудований символ **лінія**

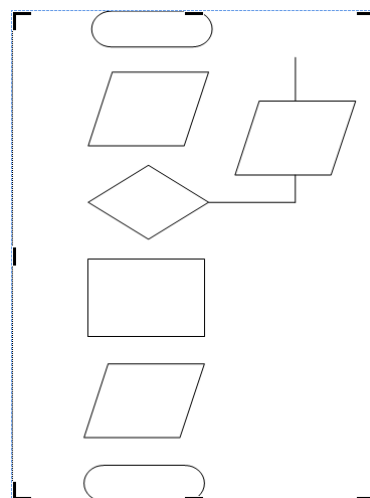


Рис. 2.16. Побудовані символи **лінія** до інших символів блок-схеми

### 2.3.9. Побудова символу блок-схеми стрілка програми

Необхідно виконати такі дії:

а) на вкладці «**Вставка**» натиснути кнопку **Фігури** (рис. 2.2);

б) у з'явившомуся списку в розділі «**Лінії**» натиснути кнопку **Стрілка** (рис. 2.17);

в) вказати курсором миші точку на середині нижнього елемента **символу блок-схеми початок та закінчення** програми – це початкова точка стрілки, а потім вказати курсором миші точку на середині верхнього елемента **символу**

**блок-схеми введення та виведення даних** – це кінцева точка стрілки (рис. 2.18).

Аналогічно побудувати **символ блок-схеми стрілка** програми до інших символів блок-схеми (рис. 2.19).

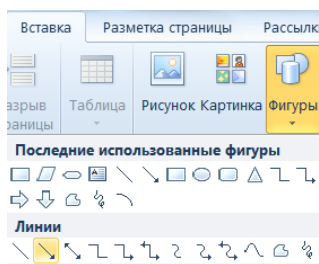


Рис. 2.17. Натиснути кнопку **Стрілка**

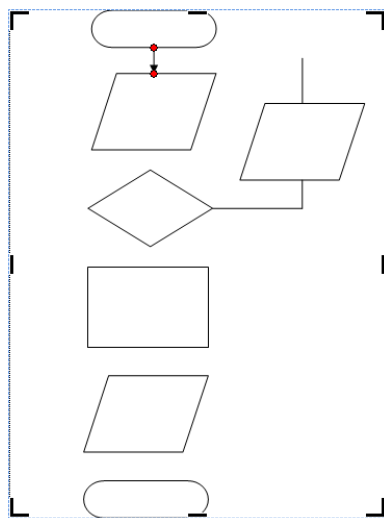


Рис. 2.18. Побудований символ **стрілка**

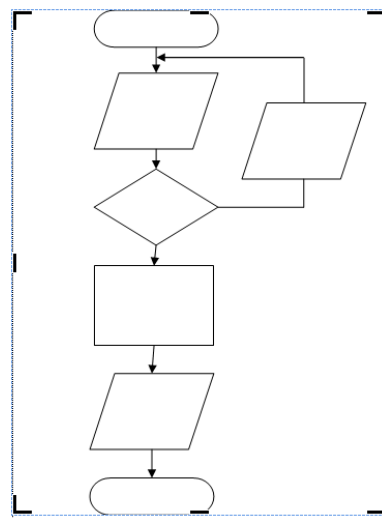


Рис. 2.19. Побудовані символи **стрілка** до інших символів блок-схеми

### 2.3.10. Заповнення текстом символів блок-схеми

Необхідно виконати такі дії:

а) виділити символ **початок та закінчення** програми, вказавши курсором миші на символі **початок та закінчення** програми (рис. 2.20);

б) натиснути праву кнопку миші на символі **початок та закінчення** програми. З'явиться контекстне меню (рис. 2.20);

в) із контекстного меню вибрати **Додати текст** (рис. 2.20);

г) у символі **початок та закінчення** з'явиться поле для вводу тексту. Ввести текст **Початок** (рис. 2.21);

д) для виходу з поля для вводу тексту в символі **початок та закінчення** необхідно вказати курсором миші у будь-якому місці документа.

Аналогічно додати текст до інших символів блок-схеми (рис. 2.22).

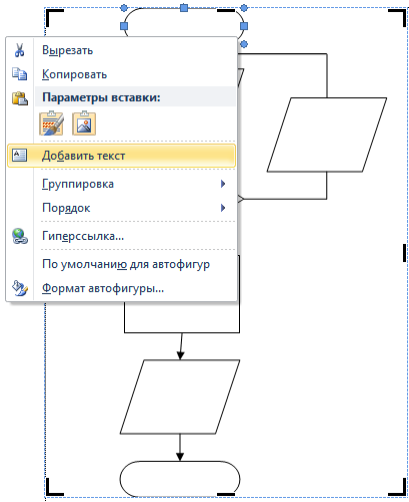


Рис. 2.20. Виділення символу початок та закінчення та вибір Додати текст

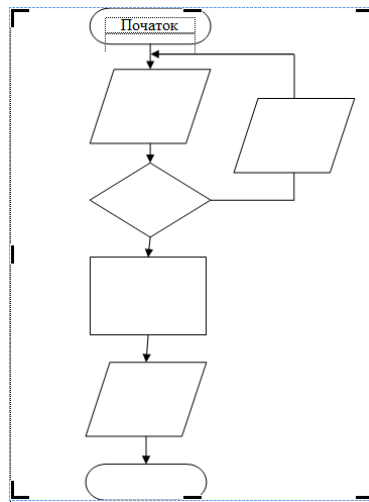


Рис. 2.21. Ввід тексту Початок

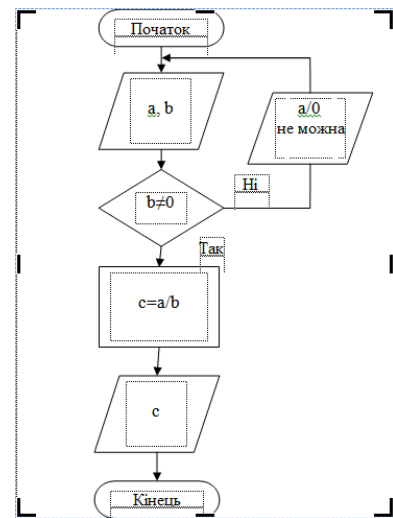


Рис. 2.22. Додавання тексту до інших символів блок-схеми

### 3. Зміст звіту з лабораторної роботи №2

**Звіт** з лабораторної роботи №2 складається з:

- титульного аркуша (див. Додаток Б);
- аркуша (див. Додаток Б) на якому розташовано: **тема** лабораторної роботи, **мета** лабораторної роботи, **складання блок-схеми** алгоритму з розгалужуючою структурою (навести **умову** та **формулу**, які розглядаються в лабораторній роботі №2, див. п. п. 2.3);
- аркуша (див. Додаток Б) на якому розташовано: **блок-схема** алгоритму з розгалужуючою структурою (навести **блок-схему**, яка розглядається в лабораторній роботі №2, див. п. п. 2.3) та **висновки**.

## Лабораторна робота №3

**Тема:** Лінійна структура програми та типи даних.

**Мета роботи:** Засвоєння початкових знань та придбання навичок, необхідних для розробки програмних продуктів комп'ютерного забезпечення в технологічних процесах обробки металів різанням.

### Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Виконати алгоритм виконання лабораторної роботи №3.
3. Скласти звіт з лабораторної роботи №3 за наведеною формою (див. Додаток В).

#### Хід виконання роботи

##### 1. Теоретичні відомості

Вивчити теоретичні відомості зі списку наведеної літератури.

##### 2. Алгоритм виконання лабораторної роботи №3

###### 2.1. Запуск програми Delphi

У системному меню «*Пуск*», операційної системи Windows, вибрати: **Пуск > Програми > Borland Delphi 7 > Delphi 7.**

###### 2.2. Створення проєкту

Після запуску програми Delphi створюється за замовчуванням проєкт типу – **Application** (Програма).

Delphi дозволяє розробляти різні програми, які в Delphi називаються проєктами. У лабораторних роботах створюються проєкти – **Application** (Програма), під операційну систему Windows з графічним інтерфейсом.

Для створення нового проєкту виберіть в «*Головному*» меню **File** (Файл) **New > Application** (Нова > Програма) (рис. 2.1).

### 2.3. Збереження проєкту


Delphi для кожного проєкту створює декілька файлів. Щоб файли різних проєктів не переплуталися між собою, слід для кожного проєкту створювати свою **папку** з іменем, яке містить назву та номер лабораторної роботи (наприклад, лабораторна робота 3 – Lab\_3).

Рекомендується імені проєкту надавати ім'я, яке містить номер лабораторної роботи (наприклад, проєкт для лабораторної роботи 3 – Project\_3).

**Файл проєкту Project\_3** потрібно зберегти в папці Lab\_3.

Необхідно також зберегти **файл модуля вихідного коду**.

Рекомендується імені файлу модуля вихідного коду надавати ім'я, як у проєкті (наприклад, Unit\_3). Його також потрібно зберегти у папці Lab\_3.

На панелі інструментів «*Standard*» (Стандартна) натиснути кнопку  **Save all** (Зберегти все) та зберегти **файли проєкту** та **файл модуля вихідного коду** згідно з вище сказаними рекомендаціями (рис. 2.2).

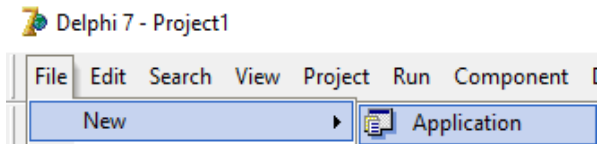


Рис. 2.1. Виберіть **New > Application** (Нова > Програма) в «*Головному*» меню **File** (Файл)

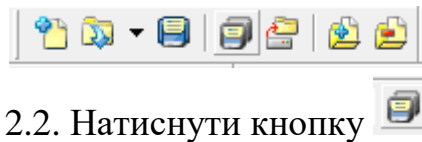


Рис. 2.2. Натиснути кнопку **Save all** (Зберегти все) на панелі інструментів «*Standard*» (Стандартна)

### 2.4. Розробка програми

Розробити програму обчислення вартості поїздки на автомобілі на дачу (туди і назад). Вихідними даними є: відстань до дачі (туди і назад) у км; кількість бензину, яку споживає автомобіль на 100 км пробігу в літрах; ціна одного літра бензину в гривнях.

Приймаємо позначення: *rast* – відстань до дачі (туди і назад) у км; *potr* – кількість бензину, яку споживає автомобіль на 100 км пробігу в літрах; *cena* – ціна одного літра бензину в гривнях.

Складаємо формулу:

$$\text{sum} = 2 \cdot (\text{potr} / 100) \cdot \text{rast} \cdot \text{cena}$$

## 2.4.1. Створення інтерфейсу програми

Інтерфейс програми представлений на рисунку 2.3. Вивід повідомлень програми представлено на рисунку 2.4 (а, б, в).

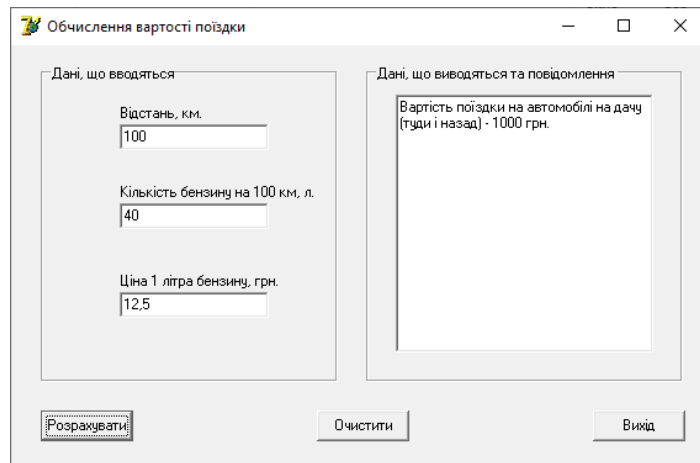
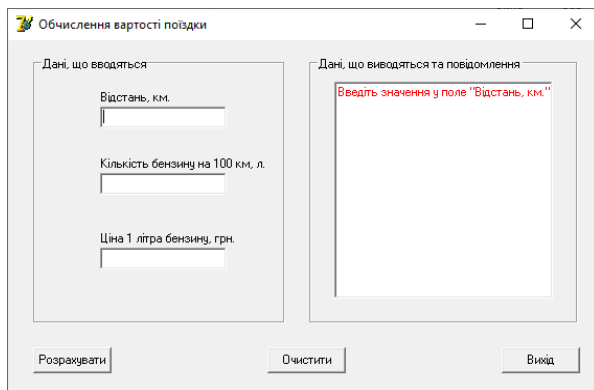
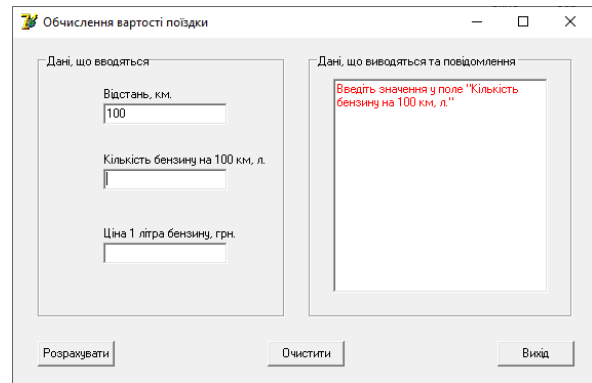


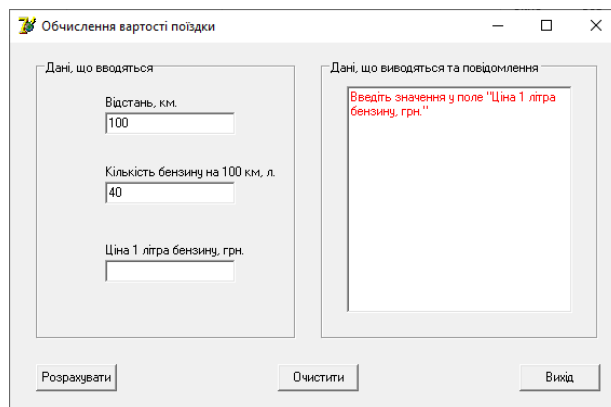
Рис. 2.3. Інтерфейс програми



а



б



в

Рис. 2.4. Вивід повідомлень програми: а, б, в.

### 2.4.1.1. Вікно форми Form1 (Форма1)

При створенні проекту типу – **Application** (Програма), Delphi автоматично створює вікно форми «**Form1**» (Форма1) (рис. 2.5). Вікно форми «**Form1**» (Форма1) являє собою шаблон вікна розробляємої програми.

### 2.4.1.2. Розміщення компонентів на вікні форми Form1 (Форма1)

На вікні форми «**Form1**» (Форма1) розмістити компоненти (рис. 2.6) вказані в таблиці 2.1.

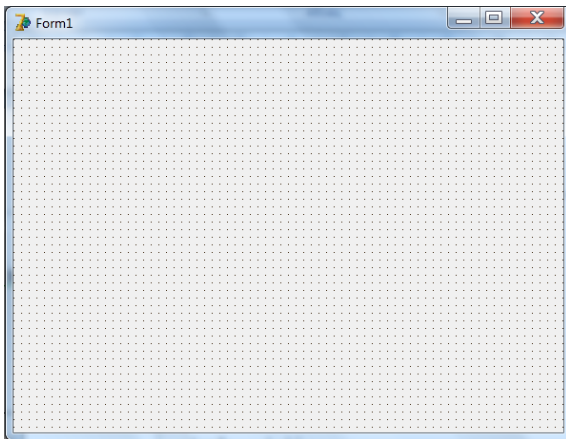


Рис. 2.5. Вікно форми «**Form1**» (Форма1)

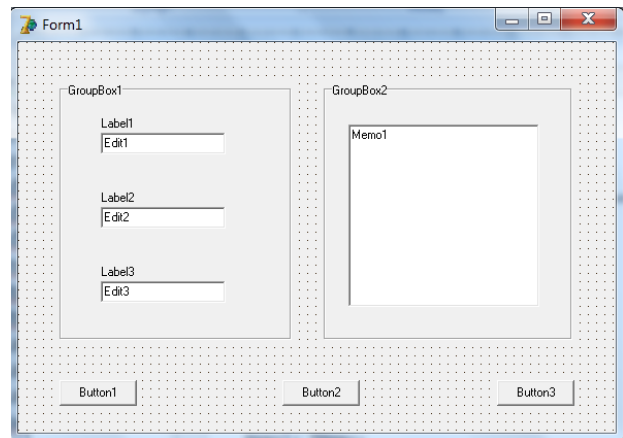


Рис. 2.6. Компоненти розміщені на вікні форми «**Form1**» (Форма1)











Для цього необхідно виконати наступні дії:

- а) на «**Панелі компонентів**» на вкладці «**Standard**» (Стандартна) (рис. 2.7) вказати курсором миші на піктограму потрібного компонента (табл. 2.1);
- б) потім вказати курсором миші у довільному місці на формі «**Form1**» (Форма1) (рис. 2.6);
- в) якщо необхідно, перетягніть компонент у потрібне місце на формі «**Form1**» (Форма1) (рис. 2.6).



Рис. 2.7. «**Панель компонентів**» на вкладці «**Standard**» (Стандартна)

Таблиця 2.1 – Компоненти та їх властивості для інтерфейсу програми

Компонент	Піктограма	Властивість	Значення
Form1	-	Caption	Обчислення вартості поїздки
		BorderStyle	bsSingle
Groupbox1		Caption	Дані, що вводяться
Groupbox2		Caption	Дані, що виводяться та повідомлення
Edit1		Name	edtInputRast
		Text	значення не вводимо, а тільки очистити поле
Edit2		Name	edtInputPotr
		Text	значення не вводимо, а тільки очистити поле
Edit3		Name	edtInputCena
		Text	значення не вводимо, а тільки очистити поле
Label1	<b>A</b>	Caption	Відстань, км.
Label2	<b>A</b>	Caption	Кількість бензину на 100 км, л.
Label3	<b>A</b>	Caption	Ціна 1 літра бензину, грн.
Memo1		Name	memOutput
		ReadOnly	True
		Lines	натиснути кнопку  . У з'явившемся вікні очистити поле, потім натиснути кнопку <b>ОК</b>
Button1		Name	btnCalculate
		Caption	Розрахувати
Button2		Name	btnClear
		Caption	Очистити
Button3		Name	btnExit
		Caption	Вихід

### 2.4.1.3. Зміна властивостей вікна форми Form1 (Форма1) та компонентів

Змінимо властивості вікна форми «*Form1*» (Форма1) та компонентів (рис. 2.8) згідно з вказаними властивостями в таблиці 2.1.

Для цього необхідно виконати наступні дії:

а) у верхній частині вікна «*Object Inspector*» (Інспектор об'єктів) (рис. 2.9) вибрати ім'я об'єкта (форми або компонентів) (табл. 2.1);

б) у лівій колонці вкладки «*Properties*» (Властивості) (рис. 2.9), де перелічені властивості об'єкта (форми чи компонентів) виділити властивість (табл. 2.1);

в) у правій колонці вкладки «*Properties*» (Властивості) (рис. 2.9), де вказано значення об'єкта (форми або компонентів) вибрати або ввести значення властивості (табл. 2.1).

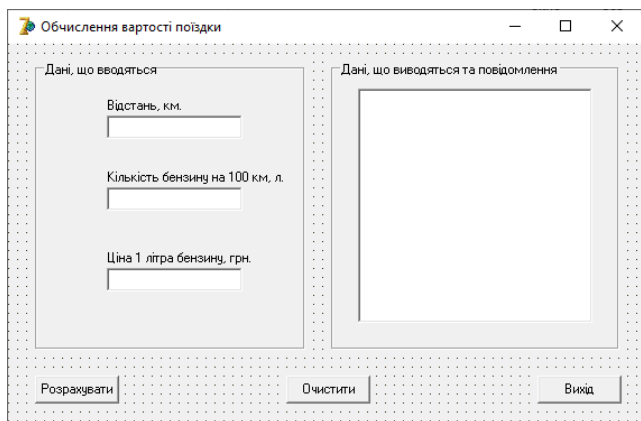


Рис. 2.8. Компоненти та вікно форми «*Form1*» (Форма1) із зміненими властивостями

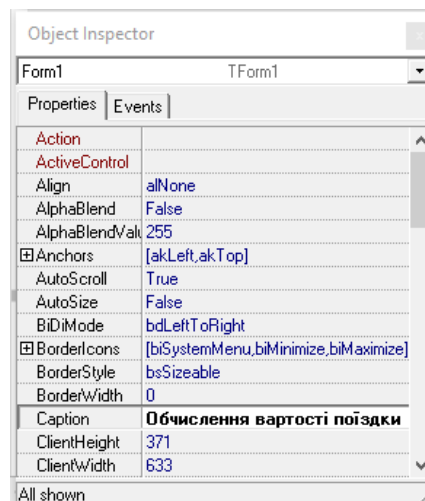


Рис. 2.9. Зміна значення властивості **Caption** (Заголовок) вікна форми «*Form1*» (Форма1) у вікні «*Object Inspector*» (Інспектор об'єктів) на **Обчислення вартості поїздки**

## 2.4.2. Створення обробників подій

Створимо обробники подій для компонентів, вказаних у таблиці 2.2.

Таблиця 2.2 – Обробники подій компонентів для програми

Компонент	Подія	Ім'я обробника
btnCalculate	OnClick	btnCalculateClick
btnClear	OnClick	btnClearClick
btnExit	OnClick	btnExitClick
edtInputRast	OnKeyPress	edtInputRastKeyPress
edtInputPotr	OnKeyPress	edtInputPotrKeyPress
edtInputCena	OnKeyPress	edtInputCenaKeyPress

Для цього необхідно виконати наступні дії:

- у верхній частині вікна «*Object Inspector*» (Інспектор об'єктів) (рис. 2.10) вибрати ім'я об'єкта (форми або компонента) (табл. 2.2);
- у цьому ж вікні потрібно вибрати вкладку «*Events*» (Події) (рис. 2.10);
- у лівій колонці вкладки «*Events*» (Події) (рис. 2.10), де перелічені імена

подій, які може використовувати вибраний об'єкт (компонент), вибрати ім'я події (табл. 2.2);

г) для створення обробника події потрібно у правій колонці вкладки «*Events*» (Події) навпроти обраної раніше події зробити подвійне клацання мишею у полі імені обробника (рис. 2.10);

д) відкриється вікно «*Unit\_3.pas*» (Редактор коду), до якого буде додано шаблон процедури обробника **btnExitClick** події **OnClick** (рис. 2.11), а у вікні «*Object Inspector*» (Інспектор об'єктів) поряд з ім'ям події з'явиться ім'я обробника (рис. 2.10);

е) у вікні «*Unit\_3.pas*» (Редактор коду) між словами **begin** і **end** необхідно ввести інструкції, що реалізують обробку події (рис. 2.11). У лістингу програми, див. п. п. 2.4.3, наведені тексти інструкцій: обробки події **OnClick** для кнопок **Розрахувати** (btnCalculate), **Очистити** (btnClear) та **Вихід** (btnExit), а також обробки події **OnKeyPress** для полів вводу edtInputRast, edtInputPotr і edtInputCena.

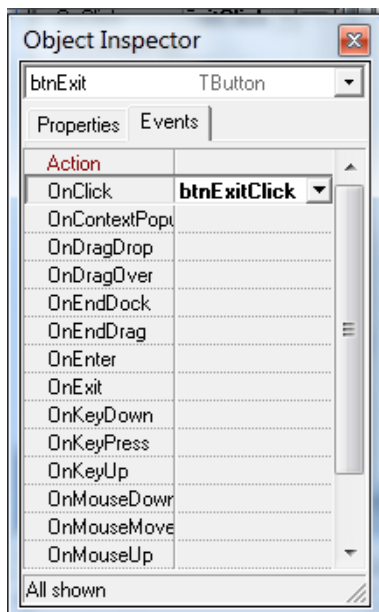


Рис. 2.10. Створення обробника **btnExitClick** події **OnClick** для компонента btnExit, на вкладці «*Events*» (Події) у вікні «*Object Inspector*» (Інспектор об'єктів)

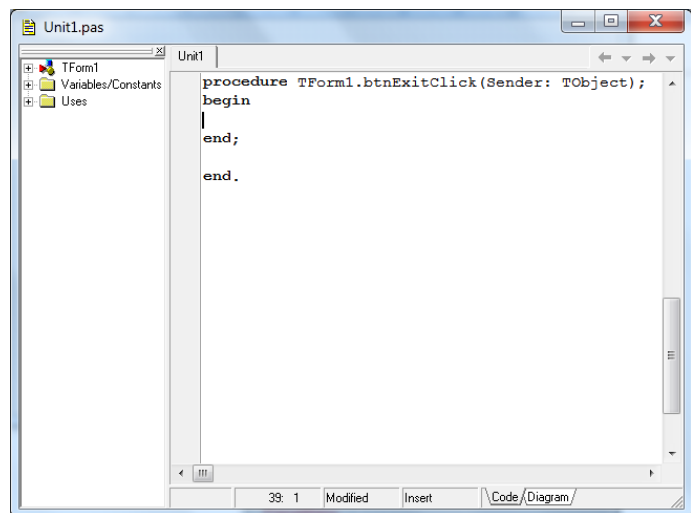


Рис. 2.11. Шаблон процедури обробника **btnExitClick** події **OnClick**, згенерований програмою Delphi

### 2.4.3. Створення вихідного коду

Вихідний код програми представлений у вигляді лістингу програми.

#### Лістинг програми

```
unit Unit_3;  
interface  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, StdCtrls;  
type  
  TForm1 = class(TForm)  
    Groupbox1: Tgroupbox;  
    Groupbox2: Tgroupbox;  
    edtInputRast: TEdit;  
    edtInputPotr: TEdit;  
    edtInputCena: TEdit;  
    Label1: TLabel;  
    Label2: TLabel;  
    Label3: TLabel;  
    memOutput: TMemo;  
    btnCalculate: TButton;  
    btnClear: TButton;  
    btnExit: TButton;  
    procedure btnCalculateClick(Sender: TObject);  
    procedure btnClearClick(Sender: TObject);  
    procedure btnExitClick(Sender: TObject);  
    procedure edtInputRastKeyPress(Sender: TObject; var Key: Char);  
    procedure edtInputPotrKeyPress(Sender: TObject; var Key: Char);  
    procedure edtInputCenaKeyPress(Sender: TObject; var Key: Char);  
private  
  { Private declarations }  
public
```

```

    { Public declarations }
end;
var
    Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1. btnCalculateClick(Sender: TObject);
const
    // Оголошення констант
    z : Integer=100;
var
    // Оголошення змінних
    Summ, Potr, Rast, Cena : Real;
    txt : string;
begin
    // Захист полів вводу
    // Відстань – Rast
    if edtInputRast.Text= ' ' then
    begin
        memOutput.Lines.Clear;
        memOutput.Font.Color:=clred;
        memOutput.Lines.Add('Введіть значення у поле «Відстань, км.»');
        edtInputRast.SetFocus;
        exit;
    end;
    // Кількість бензину на 100 км – Potr
    if edtInputPotr.Text= ' ' then
    begin
        memOutput.Lines.Clear;
        memOutput.Lines.Add('Введіть значення у поле «Кількість бензину на 100 км,
л.»');
        edtInputPotr.SetFocus;
        exit;
    end;

```

```

// Ціна 1 літра бензину – Cena
if edtInputCena.Text= ' ' then
begin
memOutput.Lines.Clear;
memOutput.Lines.Add('Введіть значення у поле «Ціна 1 літра бензину, грн.»');
edtInputCena.SetFocus;
exit;
end;
// Ввід даних
txt:=edtInputRast.Text;
Rast:=StrToFloat(txt);
txt:=edtInputPotr.Text;
Potr:=StrToFloat(txt);
txt:=edtInputCena.Text;
Cena:=StrToFloat(txt);
// Обробка даних
Summ:=2*(Potr/z)*Rast*Cena;
// Вивід даних
memOutput.font.Color:=clBlack;
memOutput.Lines.Clear;
memOutput.Lines.Add('Вартість поїздки на автомобілі на дачу '+
'(туди і назад) - '+FloatToStr(Summ)+' грн. ');
end;
procedure TForm1.BtnClearClick(Sender: TObject);
begin
// Очищення полів вводу даних
edtInputRast.Text:= ' ';
edtInputPotr.Text:= ' ';
edtInputCena. Clear;
// Очищення поля виводу даних
memOutput.Lines.Clear;
end;
procedure TForm1.BtnExitClick(Sender: TObject);
begin

```

```

// Закриття програми
Application.Terminate;
end;
// Захист поля Rast
procedure TForm1.edtInputRastKeyPress(Sender: TObject; var Key: Char);
begin
    case key of
        // Дозволено ввід цифр від 0 до 9
        '0'..'9':;
        // Дозволено ввід символу кома ", "
        ',':;
        // Заборонено ввід другої коми
        if pos(',',edtInputRast.Text)<>0 then key:=#0;
        // Дозволено використання клавіші <BackSpace> #8
        #8:;
        else
        // Ввід інших цифр та символів заборонено
        key:=#0;
        end;
end;
// Захист поля Potr
procedure TForm1.edtInputPotrKeyPress(Sender: TObject; var Key: Char);
begin
    case key of
        // Дозволено ввід цифр від 0 до 9
        '0'..'9':;
        // Дозволено ввід символу кома ", "
        ',':;
        // Заборонено ввід другої коми
        if pos(',',edtInputPotr.Text)<>0 then key:=#0;
        // Дозволено використання клавіші <BackSpace> #8
        #8:;
        else
        // Ввід інших цифр та символів заборонено

```

```

key:=#0;
end;
end;
// Захист поля Cena
procedure TForm1.edtInputCenaKeyPress(Sender: TObject; var Key: Char);
begin
  case key of
    // Дозволено ввід цифр від 0 до 9
    '0'..'9':;
    // Дозволено ввід символу кома ",",
    ',';
    // Заборонено ввід другої коми
    if pos(',',edtInputCena.Text)<>0 then key:=#0;
    // Дозволено використання клавіші <BackSpace> #8
    #8;
  else
    // Ввід інших цифр та символів заборонено
    key:=#0;
  end;
end;
end.

```

#### 2.4.4. Компіляція програми

Процес компіляції складається із двох етапів. На першому етапі виконується перевірка тексту програми на відсутність синтаксичних помилок, а на другому – генерується виконувана програма (exe-файл).

**Компіляція програми** виконується з «*Головного*» меню **Project** (Проект) обравши команду **Compile Project** (Компілювати проект) (рис. 2.12) або натиснути клавіші **Ctrl+F9**.

Повідомлення про помилки, попередження та підказки відображаються в нижній частині вікна редактора коду (рис. 2.13).

Процес усунення помилок має ітераційний характер. Зазвичай, спочатку усуваються найбільш очевидні помилки, наприклад, декларуються неоголошені

змінні. Після чергового внесення змін до тексту програми виконується повторна компіляція. Слід зважати на те, що компілятор не завжди може точно локалізувати помилку. Тому аналізуючи фрагмент програми який на думку компілятора містить помилку, потрібно звертати увагу не тільки на той фрагмент коду, на який компілятор встановив курсор, але й на той, який знаходиться у попередньому рядку.

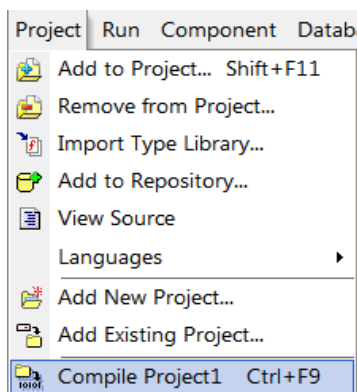


Рис. 2.12. Вибір команди **Compile Project** (Компілювати проєкт) з «Головного» меню **Project** (Проєкт)

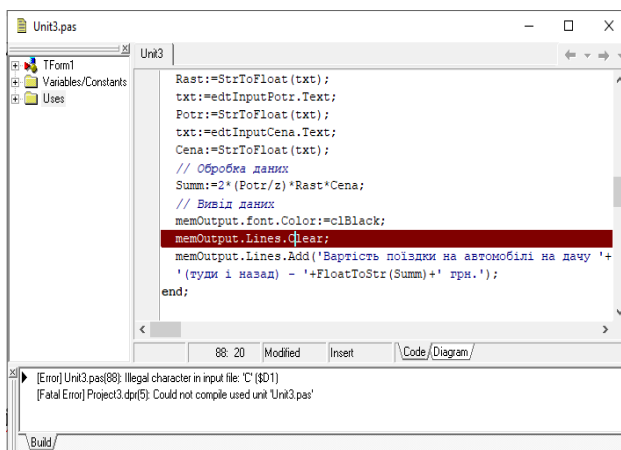


Рис. 2.13. Повідомлення компілятора про виявлені помилки

Якщо у програмі немає синтаксичних помилок, компілятор створює виконуваний файл програми. Ім'я виконуваного файлу таке ж, як і у файлу проєкту, а розширення – exe. Delphi поміщає виконуваний файл у той самий каталог, де знаходиться файл проєкту.

#### 2.4.5. Запуск програми


Запуск програми може виконуватися з «Головного» меню **Run** (Запуск) обравши **Run** (Запуск) або натиснути на кнопці  **Run** (Запуск) на панелі інструментів «**Debug**» (Налагодження) (рис. 2.14) або натиснути клавішу **F9**.



Рис. 2.14. Панель інструментів «**Debug**» (Налагодження)

## 2.4.6. Помилки часу виконання

Під час роботи програми можуть виникати помилки, які називаються **помилками часу виконання** (run-time errors) або **винятками** (exceptions). У більшості випадків причинами **винятків** є невірні вихідні дані.

Після виникнення таких помилок програма перестане робити.

Необхідно перервати неробочу програму, вибравши із «**Головного**» меню **Run** (Запуск) команду **Program Reset** (Скидання програми) (рис. 2.15).

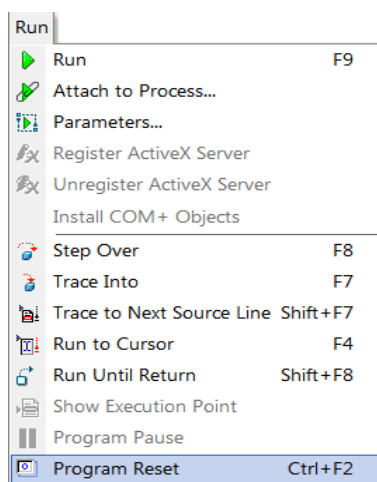


Рис. 2.15. Вибір команди **Program Reset** (Скидання програми) із «**Головного**» меню **Run** (Запуск)

## 2.4.7. Захист полів вводу

Для виключення помилок при вводі даних у поля вводу необхідно передбачити у вихідному коді програми захист полів вводу.

У лабораторній роботі №3 передбачені такі види захисту полів вводу (див. п. п. 2.4.3):

- захист полів вводу від вводу порожнього рядка;
- захист полів вводу від вводу літер та деяких символів, використовуючи подію OnKeyPress.

### 3. Зміст звіту з лабораторної роботи №3

**Звіт** з лабораторної роботи №3 складається з:

- титульного аркуша (див. Додаток В);
- аркуша (див. Додаток В) на якому розташовано: **тема** лабораторної роботи, **мета** лабораторної роботи, **розробка програми** (наведіть умову та **формулу**, які розглядаються в лабораторній роботі №3, див. п. п. 2.4);
- аркуша (див. Додаток В) на якому розташовано: **створення інтерфейсу програми** (навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №3, див. п. п. 2.4.1);
- аркушів (див. Додаток В) на яких розташовані: **лістинг програми** (навести **лістинг** програми, який розглядається в лабораторній роботі №3, див. п. п. 2.4.3) та **висновки**.

## Лабораторна робота №4

**Тема:** Розгалужуюча структура програми та типи даних.

**Мета роботи:** Засвоєння початкових знань та придбання навичок, необхідних для розробки програмних продуктів комп'ютерного забезпечення в технологічних процесах обробки металів різанням.

### Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Виконати алгоритм виконання лабораторної роботи №4.
3. Скласти звіт з лабораторної роботи №4 за наведеною формою (див. Додаток Г).

#### Хід виконання роботи

##### 1. Теоретичні відомості

Вивчити теоретичні відомості зі списку наведеної літератури.

##### 2. Алгоритм виконання лабораторної роботи №4

###### 2.1. Запуск програми Delphi

У системному меню «*Пуск*», операційної системи Windows вибрати: **Пуск > Програми > Borland Delphi 7 > Delphi 7.**

###### 2.2. Створення проєкту

Після запуску програми Delphi створюється за замовчуванням проєкт типу – **Application** (Програма). Delphi дозволяє розробляти різні програми, які в Delphi називаються проєктами. У лабораторних роботах створюються проєкти – **Application** (Програма), під операційну систему Windows з графічним інтерфейсом.

Для створення нового проєкту виберіть в «*Головному*» меню **File** (Файл) **New > Application** (Нова > Програма) (рис. 2.1).

## 2.3. Збереження проєкту

Delphi для кожного проєкту створює декілька файлів. Щоб файли різних проєктів не переплуталися між собою, слід для кожного проєкту створювати свою **папку** з іменем, яке містить назву та номер лабораторної роботи (наприклад, лабораторна робота 4 – Lab\_4).

Рекомендується імені проєкту надавати ім'я, яке містить номер лабораторної роботи (наприклад, проєкт для лабораторної роботи 4 – Project\_4).

**Файл проєкту** Project\_4 потрібно зберегти в папці Lab\_4.

Необхідно також зберегти **файл модуля вихідного коду**.

Рекомендується імені файлу модуля вихідного коду надавати ім'я, як у проєкті (наприклад, Unit\_4). Його також потрібно зберегти у папці Lab\_4.

На панелі інструментів «*Standard*» (Стандартна) натиснути кнопку **Save all** (Зберегти все) та зберегти **файли проєкту** та **файл модуля вихідного коду** згідно з вище сказаними рекомендаціями (рис. 2.2).

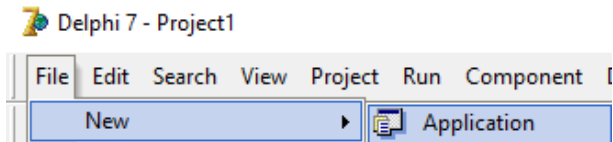


Рис. 2.1. Виберіть **New > Application** (Нова > Програма) в «*Головному*» меню **File** (Файл)

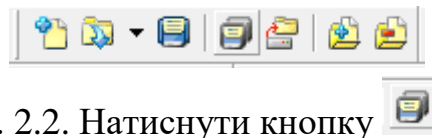


Рис. 2.2. Натиснути кнопку **Save all** (Зберегти все) на панелі інструментів «*Standard*» (Стандартна)

## 2.4. Розробка програми

Розробити програму обчислення приватного при діленні двох чисел.

Приймаємо позначення: a – ділене; b – дільник; c – приватне.

Складаємо формулу:

$$c=a/b$$

### 2.4.1. Створення інтерфейсу програми

Інтерфейс програми представлений на рисунку 2.3. Вивід повідомлень програми представлено на рисунку 2.4 (а, б, в).

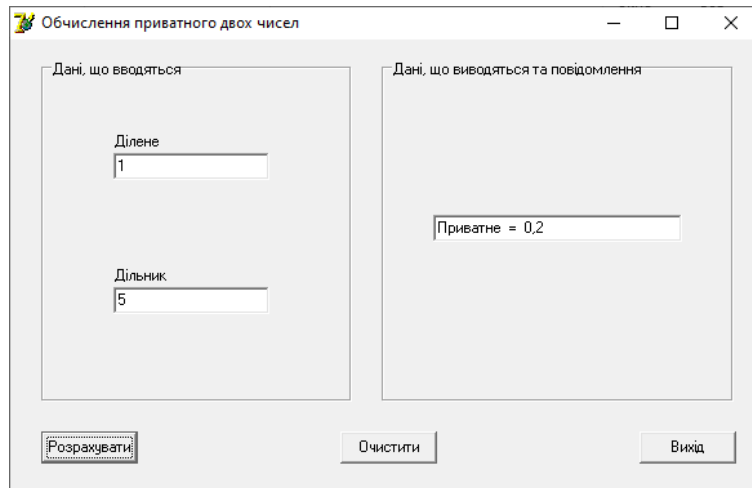
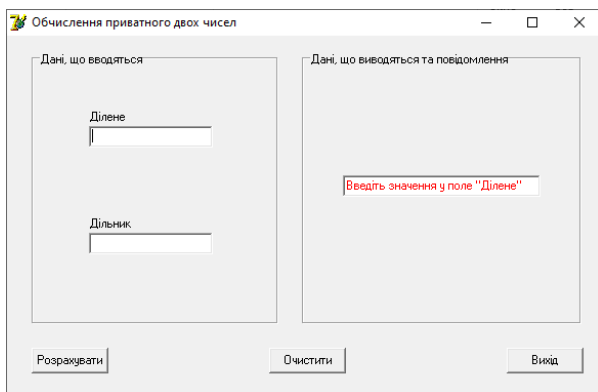
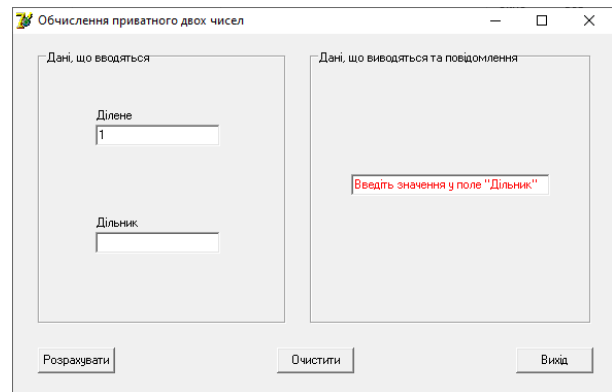


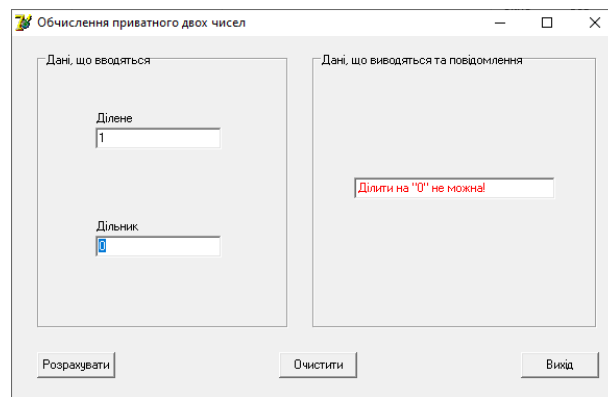
Рис. 2.3. Інтерфейс програми



а



б



в

Рис. 2.4. Вивід повідомлень програми: а, б, в.

### 2.4.1.1. Вікно форми Form1 (Форма1)

При створенні проєкту типу – **Application** (Програма), Delphi автоматично створює вікно форми «**Form1**» (Форма1) (рис. 2.5). Вікно форми «**Form1**» (Форма1) являє собою шаблон вікна розробляємої програми.

## 2.4.1.2. Розміщення компонентів на вікні форми Form1 (Форма1)

На вікні форми «*Form1*» (Форма1) розмістити компоненти (рис. 2.6) вказані в таблиці 2.1.

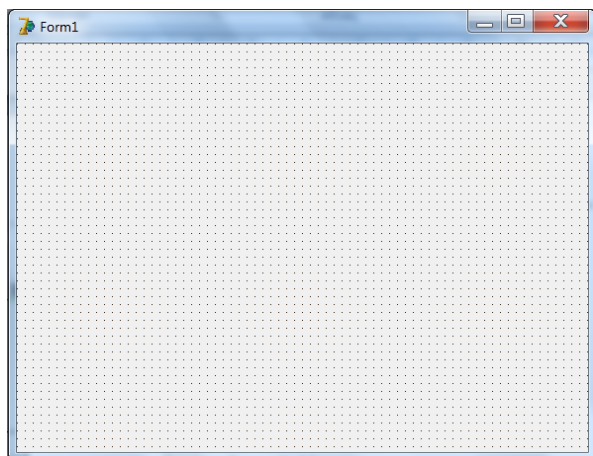


Рис. 2.5. Вікно форми «*Form1*» (Форма1)

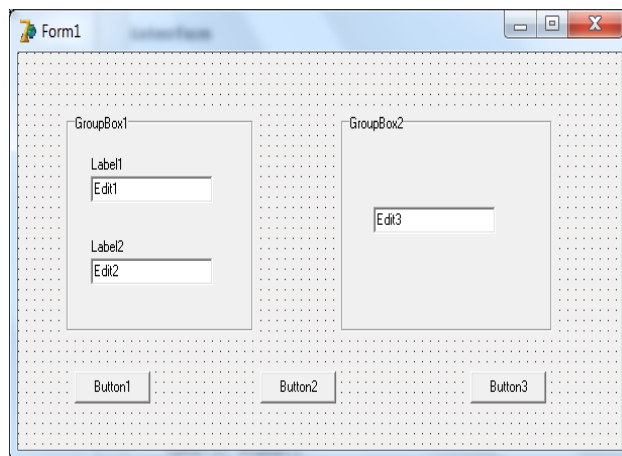


Рис. 2.6. Компоненти розміщені на вікні форми «*Form1*» (Форма1)









Для цього необхідно виконати наступні дії:

- а) на «**Панелі компонентів**» на вкладці «*Standard*» (Стандартна) (рис. 2.7) вказати курсором миші на піктограму потрібного компонента (табл. 2.1);
- б) потім вказати курсором миші у довільному місці на формі «*Form1*» (Форма1) (рис. 2.6);
- в) якщо необхідно, перетягніть компонент у потрібне місце на формі «*Form1*» (Форма1) (рис. 2.6).



Рис. 2.7. «**Панель компонентів**» на вкладці «*Standard*» (Стандартна)

Таблиця 2.1 – Компоненти та їх властивості для інтерфейсу програми

Компонент	Піктограма	Властивість	Значення
Form1	-	Caption	Обчислення приватного двох чисел
		BorderStyle	bsSingle
Groupbox1		Caption	Дані, що вводяться
Groupbox2		Caption	Дані, що виводяться та повідомлення
Edit1		Name	edtInputA
		Text	значення не вводимо, а тільки очистити поле
Edit2		Name	edtInputB
		Text	значення не вводимо, а тільки очистити поле
Edit3		Name	edtOutputC
		ReadOnly	True
		Text	значення не вводимо, а тільки очистити поле
Label1	<b>A</b>	Caption	Ділене
Label2	<b>A</b>	Caption	Дільник
Button1		Name	btnCalculate
		Caption	Розрахувати
Button2		Name	btnClear
		Caption	Очистити
Button3		Name	btnExit
		Caption	Вихід

### 2.4.1.3. Зміна властивостей вікна форми Form1 (Форма1) та компонентів

Змінимо властивості вікна форми «*Form1*» (Форма1) та компонентів (рис. 2.8) згідно з вказаними властивостями в таблиці 2.1.

Для цього необхідно виконати наступні дії:

а) у верхній частині вікна «*Object Inspector*» (Інспектор об'єктів) (рис. 2.9) вибрати ім'я об'єкта (форми або компонентів) (табл. 2.1);

б) у лівій колонці вкладки «*Properties*» (Властивості) (рис. 2.9), де перелічені властивості об'єкта (форми чи компонентів) виділити властивість (табл. 2.1);

в) у правій колонці вкладки «*Properties*» (Властивості) (рис. 2.9), де вказано значення об'єкта (форми або компонентів) вибрати або ввести значення властивості (табл. 2.1).

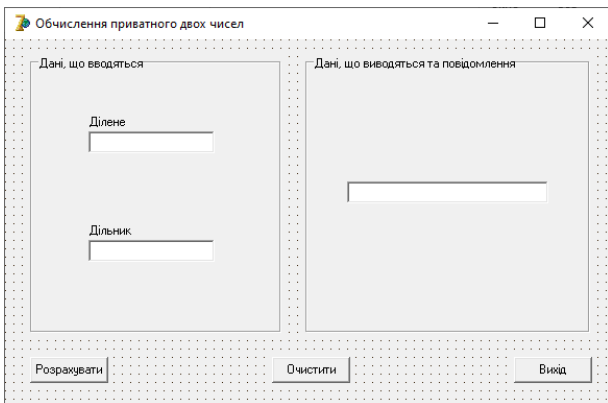


Рис. 2.8. Компоненти та вікно форми «Form1» (Форма1) із зміненими властивостями

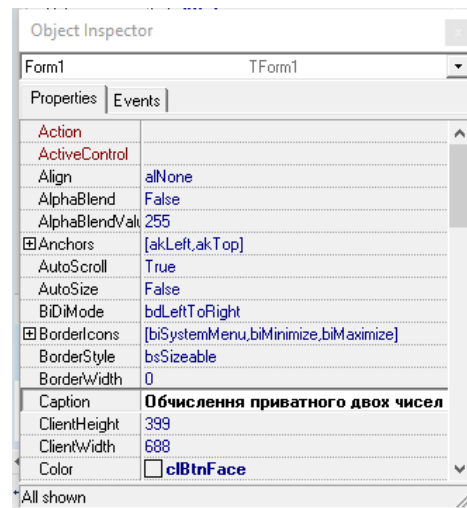


Рис. 2.9. Зміна значення властивості **Caption** (Заголовок) вікна форми «Form1» (Форма1) у вікні «Object Inspector» (Інспектор об'єктів) на **Обчислення приватного двох чисел**

### 2.4.2. Створення обробників подій

Створимо обробники подій для компонентів, вказаних у таблиці 2.2.

Таблиця 2.2 – Обробники подій компонентів для програми

Компонент	Подія	Ім'я обробника
btnCalculate	OnClick	btnCalculateClick
btnClear	OnClick	btnClearClick
btnExit	OnClick	btnExitClick
edtInputa	OnKeyPress	edtInputaKeyPress
edtInputb	OnKeyPress	edtInputbKeyPress

Для цього необхідно виконати наступні дії:

- у верхній частині вікна «Object Inspector» (Інспектор об'єктів) (рис. 2.10) вибрати ім'я об'єкта (форми або компонента) (табл. 2.2);
- у цьому ж вікні потрібно вибрати вкладку «Events» (Події) (рис. 2.10);
- у лівій колонці вкладки «Events» (Події) (рис. 2.10), де перелічені імена подій, які може використовувати вибраний об'єкт (компонент), вибрати ім'я події (табл. 2.2);
- для створення обробника події потрібно у правій колонці вкладки

«*Events*» (Події) навпроти обраної раніше події зробити подвійне клацання мишею у полі імені обробника (рис. 2.10);

д) відкриється вікно «*Unit\_4.pas*» (Редактор коду), до якого буде додано шаблон процедури обробника **btnExitClick** події **OnClick** (рис. 2.11), а у вікні «*Object Inspector*» (Інспектор об'єктів) поряд з ім'ям події з'явиться ім'я обробника (рис. 2.10);

е) у вікні «*Unit\_4.pas*» (Редактор коду) між словами **begin** і **end** необхідно ввести інструкції, що реалізують обробку події (рис. 2.11). У лістингу програми, див. п. п. 2.4.3, наведені тексти інструкцій: обробки події **OnClick** для кнопок **Розрахувати** (btnCalculate), **Очистити** (btnClear) та **Вихід** (btnExit), а також обробки події **OnKeyPress** для полів вводу edtInputa і edtInputb.

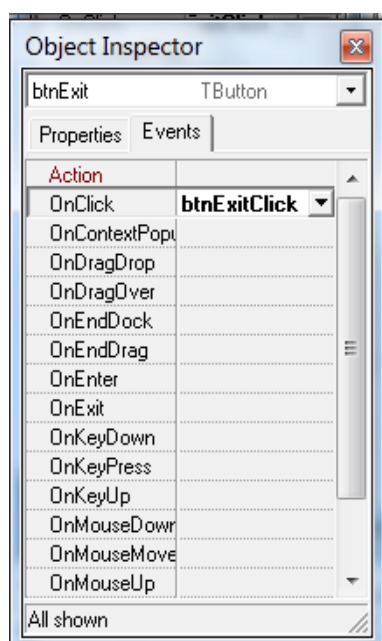


Рис. 2.10. Створення обробника **btnExitClick** події **OnClick** для компонента btnExit, на вкладці «*Events*» (Події) у вікні «*Object Inspector*» (Інспектор об'єктів)

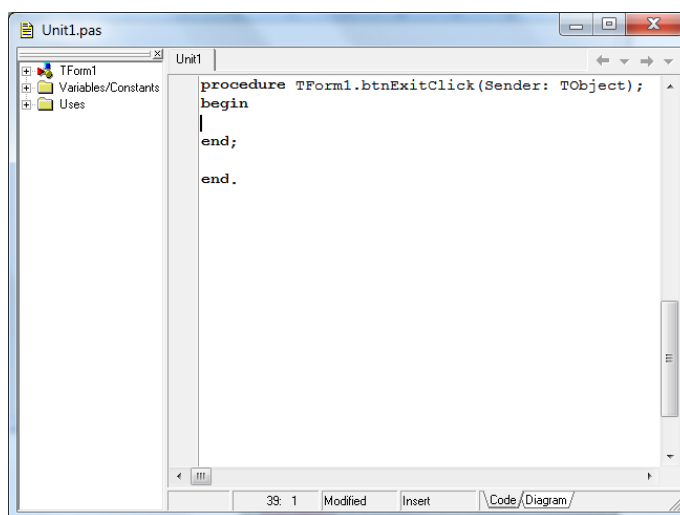


Рис. 2.11. Шаблон процедури обробника **btnExitClick** події **OnClick**, згенерований програмою Delphi

### 2.4.3. Створення вихідного коду

Вихідний код програми представлений у вигляді лістингу програми.

#### Лістинг програми

```
unit Unit_4;  
interface  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, StdCtrls;  
type  
  TForm1 = class(TForm)  
    Groupbox1: Tgroupbox;  
    Groupbox2: Tgroupbox;  
    edtInputa: TEdit;  
    edtInputb: TEdit;  
    edtOutputc: TEdit;  
    Label1: TLabel;  
    Label2: TLabel;  
    btnCalculate: TButton;  
    btnClear: TButton;  
    btnExit: TButton;  
    procedure btnCalculateClick(Sender: TObject);  
    procedure btnClearClick(Sender: TObject);  
    procedure btnExitClick(Sender: TObject);  
    procedure edtInputaKeyPress(Sender: TObject; var Key: Char);  
    procedure edtInputbKeyPress(Sender: TObject; var Key: Char);  
private  
  { Private declarations }  
public  
  { Public declarations }  
end;  
var
```

```

Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1. btnCalculateClick(Sender: TObject);
var
    // Оголошення констант
    a,b,c : Real;
    txt : String;
begin
    // Захист полів вводу
    // Ділене – a
    if edtInputa.Text= '' then
        begin
            edtOutputc.Font.Color:=clRed;
            edtOutputc.Text:= 'Введіть значення у поле «Ділене»';
            edtInputa.SetFocus;
            exit;
        end;
    // Дільник – b
    if edtInputb.Text= '' then
        begin
            edtOutputc.Clear;
            edtOutputc.Text:= 'Введіть значення у поле «Дільник»';
            edtInputb.SetFocus;
            exit;
        end;
    if edtInputb.Text='0' then
        begin
            edtOutputc.Font.Color:=clRed;
            edtOutputc.Clear;
            edtOutputc.Text:= 'Ділити на «0» не можна!';
            edtInputb.SetFocus;
            exit;
        end;

```

```

// Ввід даних
txt:=edtInputa.Text;
a:=StrToFloat(txt);
txt:=edtInputb.Text;
b:=StrToFloat(txt);
// Обробка даних
c:=a/b;
// Очищення полів виводу даних
edtOutputc.Clear;
//Вивід даних
edtOutputc.Font.Color:=clBlack;
edtOutputc.Text := 'Приватне = '+FloatToStr(c);
end;
procedure TForm1.BtnClearClick(Sender: TObject);
begin
// Очищення полів вводу даних
edtInputa.Text:= ' ';
edtInputb.Text:= ' ';
// Очищення поля виводу даних
edtOutputc.Clear;
end;
procedure TForm1.BtnExitClick(Sender: TObject);
begin
// Закриття програми
Application.Terminate;
end;
// Захист поля ділене – а
procedure TForm1.edtInputaKeyPress(Sender: TObject; var Key: Char);
begin
case key of
// Дозволено ввід цифр від 0 до 9
'0'..'9':;
// Дозволено ввід символу кома ", "
',';

```

```

// Заборонено ввід другої коми
if pos(',',edtInputa.Text)<>0 then key:=#0;
// Дозволено використання клавіші <BackSpace> #8
#8;;
else
// Ввід інших цифр та символів заборонено
key:=#0;
end;
end;
// Захист поля дільник – в
procedure TForm1.edtInputbKeyPress(Sender: TObject; var Key: Char);
begin
    case key of
        // Дозволено ввід цифр від 0 до 9
        '0'..'9':;
        // Дозволено ввід символу кома ", "
        ',':;
        // Заборонено ввід другої коми
        if pos(',',edtInputb.Text)<>0 then key:=#0;
        // Дозволено використання клавіші <BackSpace> #8
        #8;;
        else
        // Ввід інших цифр та символів заборонено
        key:=#0;
        end;
    end;
end.

```

#### 2.4.4. Компіляція програми

Процес компіляції складається із двох етапів. На першому етапі виконується перевірка тексту програми на відсутність синтаксичних помилок, а на другому – генерується виконувана програма (exe-файл).

**Компіляція програми** виконується з «*Головного*» меню **Project** (Проект)

обравши команду **Compile Project** (Компілювати проєкт) (рис. 2.12) або натиснути клавіші **Ctrl+F9**.

Повідомлення про помилки, попередження та підказки відображаються в нижній частині вікна редактора коду (рис. 2.13).

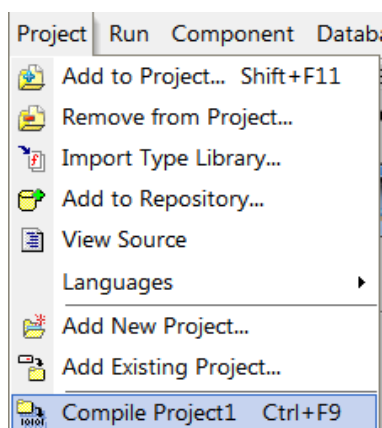


Рис. 2.12. Вибір команди **Compile Project** (Компілювати проєкт) з «Головного» меню **Project** (Проект)

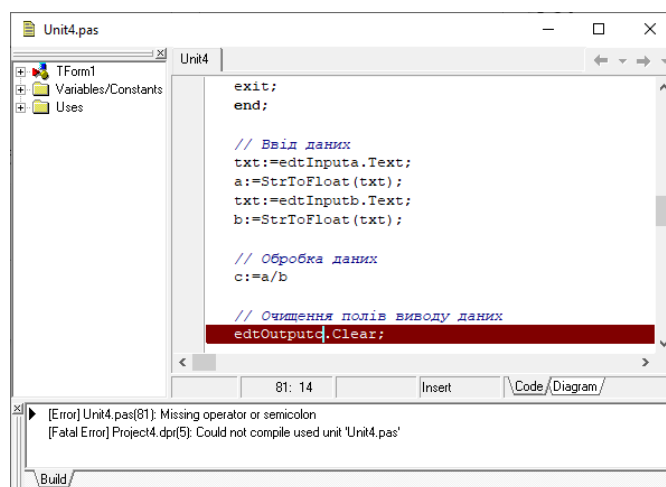



Рис. 2.13. Повідомлення компілятора про виявлені помилки

Процес усунення помилок має ітераційний характер. Зазвичай, спочатку усуваються найбільш очевидні помилки, наприклад, декларуються неоголошені змінні. Після чергового внесення змін до тексту програми виконується повторна компіляція. Слід зважати на те, що компілятор не завжди може точно локалізувати помилку. Тому аналізуючи фрагмент програми який на думку компілятора містить помилку, потрібно звертати увагу не тільки на той фрагмент коду, на який компілятор встановив курсор, але й на той, який знаходиться у попередньому рядку.

Якщо у програмі немає синтаксичних помилок, компілятор створює виконуваний файл програми. Ім'я виконуваного файлу таке ж, як і у файлу проєкту, а розширення – exe. Delphi поміщає виконуваний файл у той самий каталог, де знаходиться файл проєкту.

#### 2.4.5. Запуск програми

Запуск програми може виконуватися з «Головного» меню **Run** (Запуск) обравши **Run** (Запуск) або натиснути на кнопки  **Run** (Запуск) на панелі

інструментів «*Debug*» (Налагодження) (рис. 2.14) або натиснути клавішу **F9**.

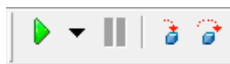


Рис. 2.14. Панель інструментів «*Debug*» (Налагодження)

### 2.4.6. Помилки часу виконання

Під час роботи програми можуть виникати помилки, які називаються **помилками часу виконання** (run-time errors) або **винятками** (exceptions). У більшості випадків причинами **винятків** є невірні вихідні дані.

Після виникнення таких помилок програма перестане робити.

Необхідно перервати неробочу програму, вибравши із «*Головного*» меню **Run** (Запуск) команду **Program Reset** (Скидання програми) (рис. 2.15).

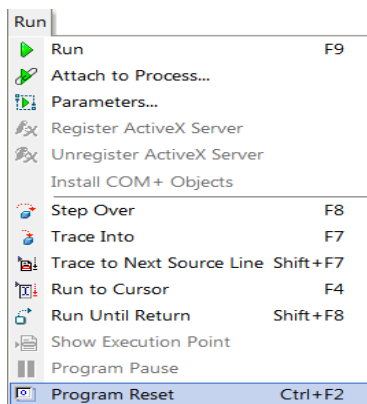


Рис. 2.15. Вибір команди **Program Reset** (Скидання програми) із «*Головного*» меню **Run** (Запуск)

### 2.4.7 Захист полів вводу

Для виключення помилок, при вводі даних у поля вводу необхідно передбачити у вихідному коді програми захист полів вводу.

У лабораторній роботі №4 передбачені такі види захисту полів вводу (див. п. п. 2.4.3):

- захист полів вводу від вводу порожнього рядка;
- захист поля вводу дільник від вводу цифри «0» нуль;
- захист полів вводу від вводу літер та деяких символів, використовуючи подію OnKeyPress.

### 3. Зміст звіту з лабораторної роботи №4

**Звіт** з лабораторної роботи №4 складається з:

- титульного аркуша (див. Додаток Г);
- аркуша (див. Додаток Г) на якому розташовано: **тема** лабораторної роботи, **мета** лабораторної роботи, **розробка програми** (наведіть умову та формулу, які розглядаються в лабораторній роботі №4, див. п. п. 2.4);
- аркуша (див. Додаток Г) на якому розташовано: **створення інтерфейсу програми** (навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №4, див. п. п. 2.4.1);
- аркушів (див. Додаток Г) на яких розташовані: **лістинг програми** (навести **лістинг** програми, який розглядається в лабораторній роботі №4, див. п. п. 2.4.3) та **висновки**.

#### Запитання з самоконтролю до розділу 1

1. Що таке компілятор?
2. Що таке вихідний код?
3. Що таке мова високого рівня?
4. Що таке алгоритм?
5. Що таке блок-схема?
6. Які типи даних використовуються при програмуванні на мові Object Pascal?
7. Що таке змінна?
8. Що таке константа?

## РОЗДІЛ 2. МЕТОДИ ВВОДУ ДАНИХ

### Лабораторна робота №5

**Тема:** Математичні функції та процедури.

**Мета роботи:** Засвоєння початкових знань та придбання навичок, необхідних для розробки програмних продуктів комп'ютерного забезпечення в технологічних процесах обробки металів різанням.

### Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Виконати алгоритм виконання лабораторної роботи №5.
3. Скласти звіт з лабораторної роботи №5 за наведеною формою (див. Додаток Д).

#### Хід виконання роботи

##### 1. Теоретичні відомості

Вивчити теоретичні відомості зі списку наведеної літератури.

##### 2. Алгоритм виконання лабораторної роботи №5

###### 2.1. Запуск програми Delphi

У системному меню «*Пуск*», операційної системи Windows вибрати: **Пуск > Програми > Borland Delphi 7 > Delphi 7.**

###### 2.2. Створення проекту

Після запуску програми Delphi створюється за замовчуванням проект типу – **Application** (Програма). Delphi дозволяє розробляти різні програми, які в Delphi називаються проектами. У лабораторних роботах створюються проекти – **Application** (Програма), під операційну систему Windows з графічним інтерфейсом.

Для створення нового проєкту виберіть в «*Головному*» меню **File** (Файл) **New > Application** (Нова > Програма) (рис. 2.1).

### 2.3. Збереження проєкту

Delphi для кожного проєкту створює декілька файлів. Щоб файли різних проєктів не переплуталися між собою, слід для кожного проєкту створювати свою **папку** з іменем, яке містить назву та номер лабораторної роботи (наприклад, лабораторна робота 5 – Lab\_5).

Рекомендується імені проєкту надавати ім'я, яке містить номер лабораторної роботи (наприклад, проєкт для лабораторної роботи 5 – Project\_5).

**Файл проєкту Project\_5** потрібно зберегти в папці Lab\_5.

Необхідно також зберегти **файл модуля вихідного коду**.

Рекомендується імені файлу модуля вихідного коду надавати ім'я, як у проєкті (наприклад, Unit\_5). Його також потрібно зберегти у папці Lab\_5.

На панелі інструментів «*Standard*» (Стандартна) натиснути кнопку **Save all** (Зберегти все) та зберегти **файли проєкту** та **файл модуля вихідного коду** згідно з вище сказаними рекомендаціями (рис. 2.2).

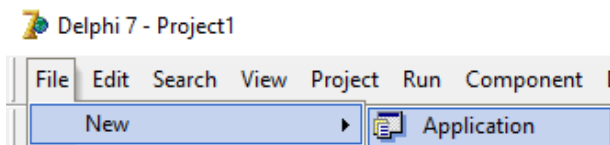


Рис. 2.1. Виберіть **New > Application** (Нова > Програма) в «*Головному*» меню **File** (Файл)

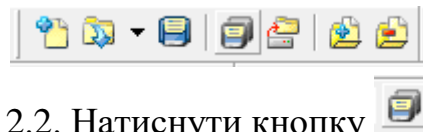


Рис. 2.2. Натиснути кнопку **Save all** (Зберегти все) на панелі інструментів «*Standard*» (Стандартна)

### 2.4. Розробка програми

Розробити програму обчислення значень змінних *a* та *b*, якщо:  $x = 2,1$ ;  $y = 0,59$  та  $z = -4,8$ , використовуючи формули:

$$a = y \cdot \operatorname{tg}^3(x^2) + \sqrt{\frac{z^2}{y^2 + x^2}}$$

$$b = \ln(y + x^2) + \sin^2\left(\frac{z}{x}\right)$$

## 2.4.1. Створення інтерфейсу програми

Інтерфейс програми представлений на рисунку 2.3. Вивід повідомлень програми представлено на рисунку 2.4 (а, б, в, г, д).

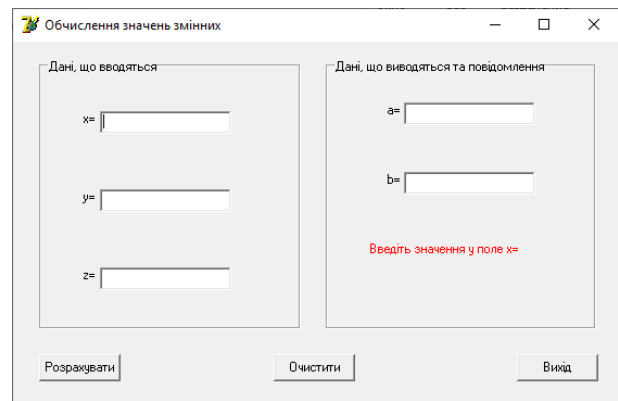
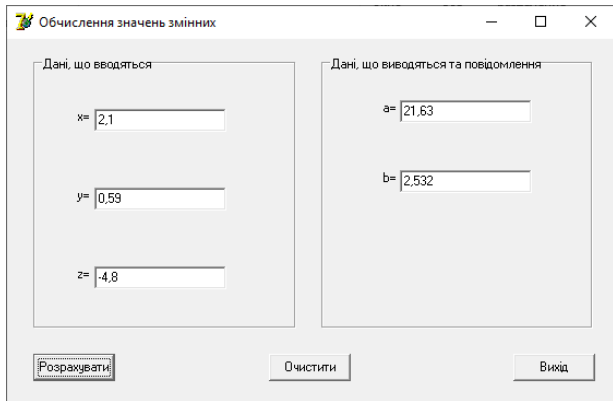
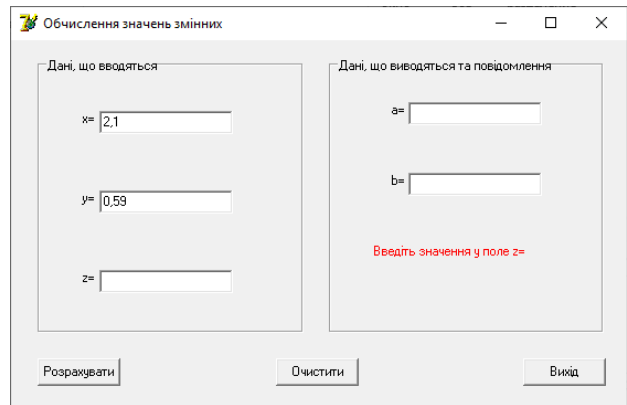
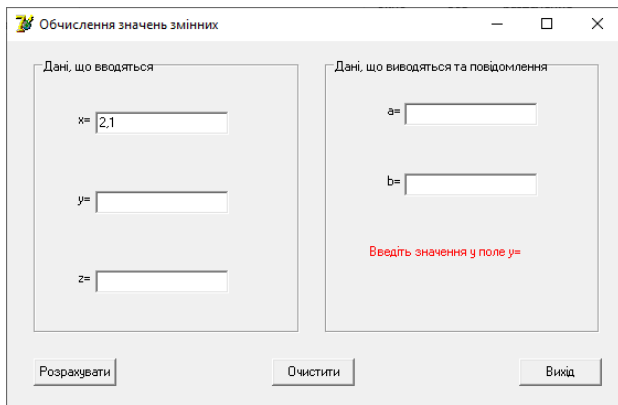


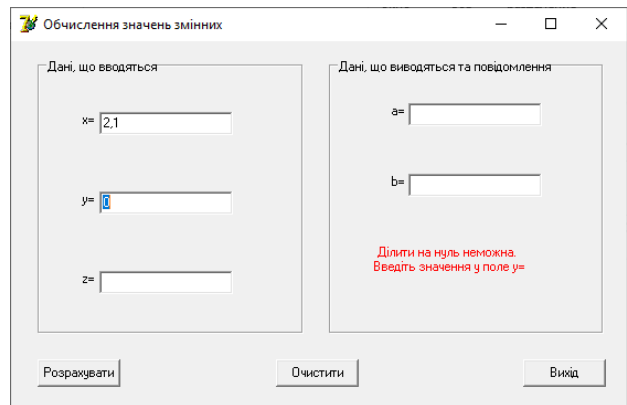
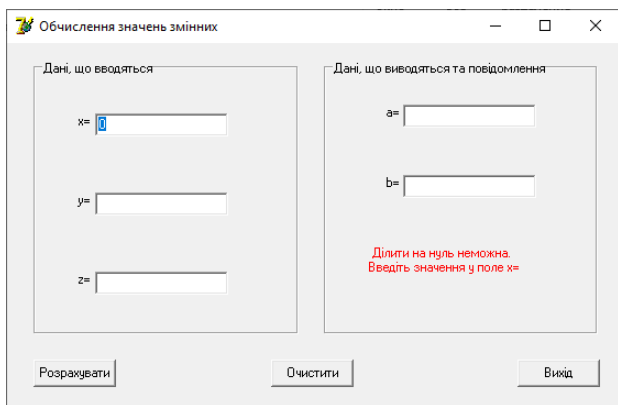
Рис. 2.3. Інтерфейс програми

а



б

в



г

д

Рис. 2.4. Вивід повідомлень програми: а, б, в, г, д.

### 2.4.1.1. Вікно форми Form1 (Форма1)

При створенні проекту типу – **Application** (Програма), Delphi автоматично створює вікно форми «**Form1**» (Форма1) (рис. 2.5). Вікно форми «**Form1**» (Форма1) являє собою шаблон вікна розробляємої програми.

### 2.4.1.2. Розміщення компонентів на вікні форми Form1 (Форма1)

На вікні форми «**Form1**» (Форма1) розмістити компоненти (рис. 2.6) вказані в таблиці 2.1.

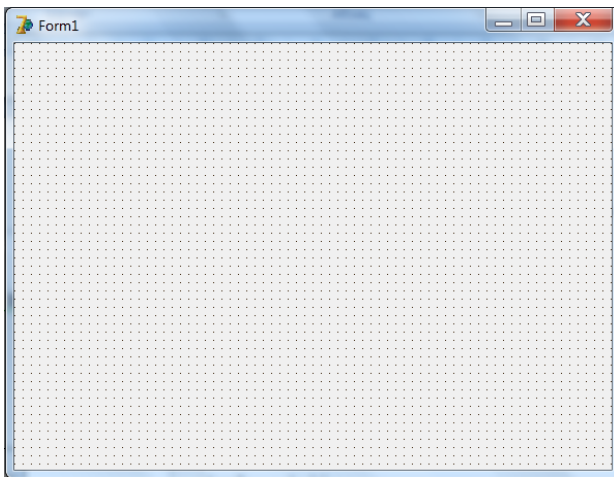


Рис. 2.5. Вікно форми «**Form1**» (Форма1)

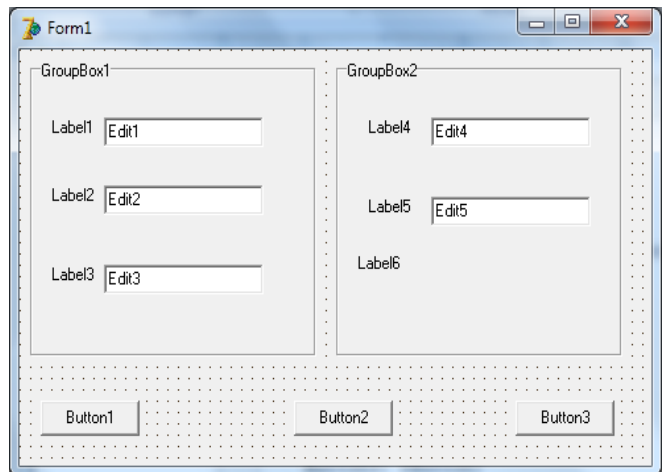


Рис. 2.6. Компоненти розміщені на вікні форми «**Form1**» (Форма1)

Для цього необхідно виконати наступні дії:

а) на «**Панелі компонентів**» на вкладці «**Standard**» (Стандартна) (рис. 2.7), вказати курсором миші на піктограму потрібного компонента (табл. 2.1);

б) потім вказати курсором миші у довільному місці на формі «**Form1**» (Форма1) (рис. 2.6);

в) якщо необхідно, перетягніть компонент у потрібне місце на формі «**Form1**» (Форма1) (рис. 2.6).

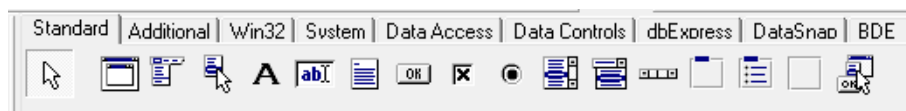












Рис. 2.7. «**Панель компонентів**» на вкладці «**Standard**» (Стандартна)

Таблиця 2.1 – Компоненти та їх властивості для інтерфейсу програми

Компонент	Піктограма	Властивість	Значення
Form1	-	Caption	Обчислення значень змінних
		BorderStyle	bsSingle
GroupBox1		Caption	Дані, що вводяться
GroupBox2		Caption	Дані, що виводяться та повідомлення
Edit1		Name	edtInputx
		Text	значення не вводимо, а тільки очистити поле
Edit2		Name	edtInputy
		Text	значення не вводимо, а тільки очистити поле
Edit3		Name	edtInputz
		Text	значення не вводимо, а тільки очистити поле
Edit4		Name	edtOutputa
		ReadOnly	True
		Text	значення не вводимо, а тільки очистити поле
Edit5		Name	edtOutputb
		ReadOnly	True
		Text	значення не вводимо, а тільки очистити поле
Label1	<b>A</b>	Caption	x=
Label2	<b>A</b>	Caption	y=
Label3	<b>A</b>	Caption	z=
Label4	<b>A</b>	Caption	a=
Label5	<b>A</b>	Caption	b=
Label6	<b>A</b>	Name	lblOutputms
		Caption	значення не вводимо, а тільки очистити поле
Button1		Name	btnCalculate
		Caption	Розрахувати
Button2		Name	btnClear
		Caption	Очистити
Button3		Name	btnExit
		Caption	Вихід

### 2.4.1.3. Зміна властивостей вікна форми Form1 (Форма1) та компонентів

Змінимо властивості вікна форми «*Form1*» (Форма1) та компонентів (рис. 2.8) згідно з вказаними властивостями в таблиці 2.1.

Для цього необхідно виконати наступні дії:

а) у верхній частині вікна «*Object Inspector*» (Інспектор об'єктів) (рис. 2.9) вибрати ім'я об'єкта (форми або компонентів) (табл. 2.1);

б) у лівій колонці вкладки «*Properties*» (Властивості) (рис. 2.9), де перелічені властивості об'єкта (форми чи компонентів) виділити властивість (табл. 2.1);

в) у правій колонці вкладки «*Properties*» (Властивості) (рис. 2.9), де вказано значення об'єкта (форми або компонентів) вибрати або ввести значення властивості (табл. 2.1).

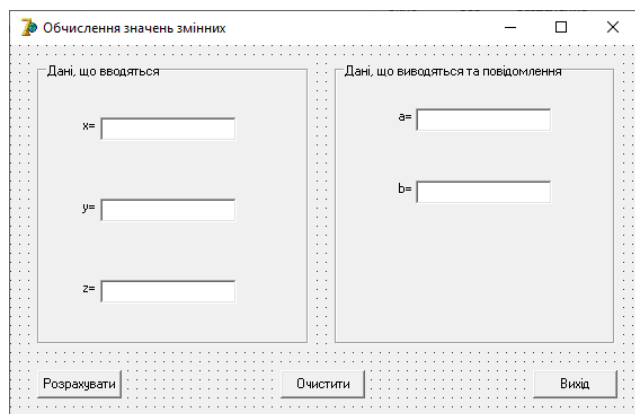


Рис.2.8. Компоненти та вікно форми «*Form1*» (Форма1) із зміненими властивостями

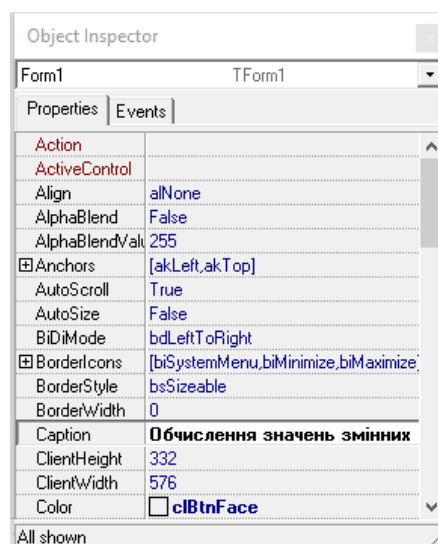


Рис.2.9. Зміна значення властивості **Caption** (Заголовок) вікна форми «*Form1*» (Форма1) у вікні «*Object Inspector*» (Інспектор об'єктів) на **Обчислення значень змінних**

## 2.4.2. Створення обробників подій

Створимо обробники подій для компонентів, вказаних у таблиці 2.2.

Таблиця 2.2 – Обробники подій компонентів для програми

Компонент	Подія	Ім'я обробника
btnCalculate	OnClick	btnCalculateClick
btnClear	OnClick	btnClearClick
btnExit	OnClick	btnExitClick
edtInputx	OnKeyPress	edtInputxKeyPress
edtInputy	OnKeyPress	edtInputyKeyPress
edtInputz	OnKeyPress	edtInputzKeyPress

Для цього необхідно виконати наступні дії:

- а) у верхній частині вікна «*Object Inspector*» (Інспектор об'єктів) (рис. 2.10) вибрати ім'я об'єкта (форми або компонента) (табл. 2.2);
- б) у цьому ж вікні потрібно вибрати вкладку «*Events*» (Події) (рис. 2.10);
- в) у лівій колонці вкладки «*Events*» (Події) (рис. 2.10), де перелічені імена подій, які може використовувати вибраний об'єкт (компонент), вибрати ім'я події (табл. 2.2);
- г) для створення обробника події потрібно у правій колонці вкладки «*Events*» (Події) навпроти обраної раніше події зробити подвійне клацання мишею у полі імені обробника (рис. 2.10);
- д) відкриється вікно «*Unit\_5.pas*» (Редактор коду), до якого буде додано шаблон процедури обробника **btnExitClick** події **OnClick** (рис. 2.11), а у вікні «*Object Inspector*» (Інспектор об'єктів) поряд з ім'ям події з'явиться ім'я обробника (рис. 2.10);
- е) у вікні «*Unit\_5.pas*» (Редактор коду) між словами **begin** і **end** необхідно ввести інструкції, що реалізують обробку події (рис. 2.11). У лістингу програми, див. п. п. 2.4.3, наведені тексти інструкцій: обробки події **OnClick** для кнопок **Розрахувати** (btnCalculate), **Очистити** (btnClear) та **Вихід** (btnExit), а також обробки події **OnKeyPress** для полів вводу edtInputx, edtInputy і edtInputz.

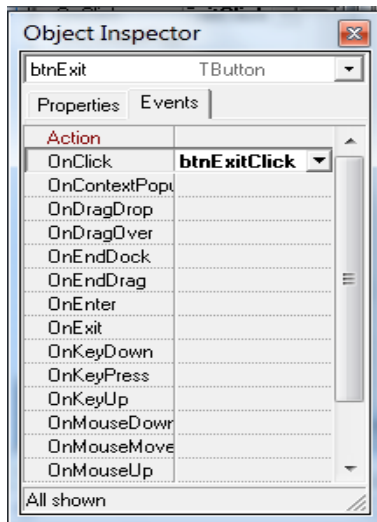


Рис.2.10. Створення обробника **btnExitClick** події **OnClick** для компонента **btnExit**, на вкладці «*Events*» (Події) у вікні «*Object Inspector*» (Інспектор об'єктів)

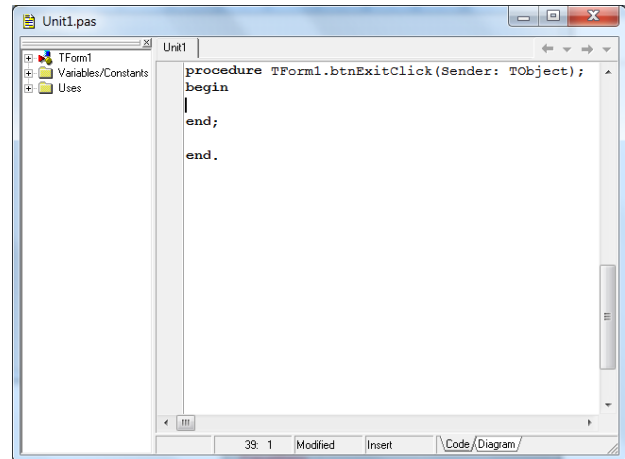


Рис.2.11. Шаблон процедури обробника **btnExitClick** події **OnClick**, згенерований програмою Delphi

### 2.4.3. Створення вихідного коду

Вихідний код програми представлений у вигляді лістингу програми.

Так як у вихідному коді використовуються математичні функції, то необхідно в розділі **uses** додати модуль **Math**.

#### Лістинг програми

```

unit Unit_5;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Math;
type
  TForm1 = class(TForm)
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    edtInputx: TEdit;
    edtInputy: TEdit;

```

```

edtInputz: TEdit;
edtOutputa: TEdit;
edtOutputb: TEdit;
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
btnCalculate: TButton;
btnClear: TButton;
btnExit: TButton;
procedure btnCalculateClick(Sender: TObject);
procedure btnClearClick(Sender: TObject);
procedure btnExitClick(Sender: TObject);
procedure edtInputxKeyPress(Sender: TObject; var Key: Char);
procedure edtInputyKeyPress(Sender: TObject; var Key: Char);
procedure edtInputzKeyPress(Sender: TObject; var Key: Char);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
implementation
  {$R *.dfm}
procedure TForm1.btnCalculateClick(Sender: TObject);
var
  // Оголошення змінних
  txt : String;
  x, y, z, a, b : Real;
begin
  // Захист полів вводу
  // x

```

```

if edtInputx.Text = ' ' then
begin
  lblOutputms.Caption:=' ';
  lblOutputms.Font.Color:=clred;
  lblOutputms.Caption:= 'Введіть значення у поле x=';
  edtInputx.SetFocus;
  exit;
end;
if edtInputx.Text = '0' then
begin
  lblOutputms.Caption:=' ';
  lblOutputms.Font.Color:=clred;
  lblOutputms.Caption:= 'Ділити на нуль неможна.'+#13#10 +'Введіть значення у
поле x=';
  edtInputx.SetFocus;
  exit;
end;
// y
if edtInputy.Text= ' ' then
begin
  lblOutputms.Caption:=' ';
  lblOutputms.Font.Color:=clred;
  lblOutputms.Caption:= 'Введіть значення у поле y=';
  edtInputy.SetFocus;
  exit;
end;
if edtInputy.Text= '0' then
begin
  lblOutputms.Caption:=' ';
  lblOutputms.Font.Color:=clred;
  lblOutputms.Caption:= ' Ділити на нуль неможна.'+#13#10 +'Введіть значення у
поле y=';
  edtInputy.SetFocus;
  exit;

```

```

end;
// z
if edtInputz.Text= ' ' then
begin
lblOutputms.Caption:=' ';
lblOutputms.Font.Color:=clred;
lblOutputms.Caption:= 'Введіть значення у поле z=';
edtInputz.SetFocus;
exit;
end;
// Очищення поля повідомлення
lblOutputms.Caption:=' ';
// Ввід даних
txt := edtInputx.Text;
x := StrToFloat(txt);
txt := edtInputy.Text;
y := StrToFloat(txt);
txt := edtInputz.Text;
z := StrToFloat(txt);
// Обробка даних
a:=(y*power(tan(sqrt(x)),3))+sqrt(sqrt(z)/(sqrt(y)+sqrt(x))));
b:=ln(y+sqrt(x))+sqrt(sin(z/y));
// Вивід даних з форматуванням
edtOutputa.Text := FloatToStrF(a,ffGeneral,4,2);
edtOutputb.Text := FloatToStrF(b,ffGeneral,4,2);
end;
procedure TForm1.btnExitClick(Sender: TObject);
begin
// Закриття програми
Application.Terminate;
end;
procedure TForm1.btnClearClick(Sender: TObject);
begin
// Очищення полів вводу даних

```

```

edtInputx.Text:= ' ';
edtInputy.Text:= ' ';
edtInputz.Text:= ' ';
// Очищення полів виводу даних
edtOutputa.Clear;
edtOutputb.Clear;
// Очищення поля повідомлення
lblOutputms.Caption:= ' ';
end;
// Захист поля x
procedure TForm1.edtInputxKeyPress(Sender: TObject; var Key: Char);
begin
    case key of
        // Дозволено ввід цифр від 0 до 9
        '0'..'9':;
        // Дозволено ввід символу кома ", "
        ',':;
        // Заборонено ввід другої коми
        if pos(',',edtInputx.Text)<>0 then key:=#0;
        // Дозволено використання клавіші <BackSpace> #8
        #8:;
        // Дозволено введення символу мінус "-"
        '-':;
        // Заборонено введення другого мінусу
        if pos('-',edtInputx.Text)<>0 then key:=#0;
        else
        // Ввід інших цифр та символів заборонено
        key:=#0;
    end;
end;
// Захист поля y
procedure TForm1.edtInputyKeyPress(Sender: TObject; var Key: Char);
begin
    case key of

```

```

// Дозволено ввід цифр від 0 до 9
'0'..'9':;
// Дозволено ввід символу кома ",",
':;
// Заборонено ввід другої коми
if pos(',',edtInputy.Text)<>0 then key:=#0;
// Дозволено використання клавіші <BackSpace> #8
#8:;
// Дозволено введення символу мінус "-"
'-':
// Заборонено введення другого мінусу
if pos('-',edtInputy.Text)<>0 then key:=#0;
else
// Ввід інших цифр та символів заборонено
key:=#0;
end;
end;
// Захист поля z
procedure TForm1.edtInputzKeyPress(Sender: TObject; var Key: Char);
begin
  case key of
    // Дозволено ввід цифр від 0 до 9
    '0'..'9':;
    // Дозволено ввід символу кома ",",
    ':;
    // Заборонено ввід другої коми
    if pos(',',edtInputz.Text)<>0 then key:=#0;
    // Дозволено використання клавіші <BackSpace> #8
    #8:;
    // Дозволено введення символу мінус "-"
    '-':
    // Заборонено введення другого мінусу
    if pos('-',edtInputz.Text)<>0 then key:=#0;
    else

```

```
// Ввід інших цифр та символів заборонено
key:=#0;
end;
end;
end.
```

#### 2.4.4. Компіляція програми

Процес компіляції складається із двох етапів. На першому етапі виконується перевірка тексту програми на відсутність синтаксичних помилок, а на другому – генерується виконувана програма (exe-файл).

**Компіляція програми** виконується з «*Головного*» меню **Project** (Проект) обравши команду **Compile Project** (Компілювати проєкт) (рис. 2.12) або натиснути клавіші **Ctrl+F9**.

Повідомлення про помилки, попередження та підказки відображаються в нижній частині вікна редактора коду (рис. 2.13).

Процес усунення помилок має ітераційний характер. Зазвичай, спочатку усуваються найбільш очевидні помилки, наприклад, декларуються неоголошені змінні. Після чергового внесення змін до тексту програми виконується повторна компіляція. Слід зважати на те, що компілятор не завжди може точно локалізувати помилку. Тому аналізуючи фрагмент програми який на думку компілятора містить помилку, потрібно звертати увагу не тільки на той фрагмент коду, на який компілятор встановив курсор, але й на той, який знаходиться у попередньому рядку.

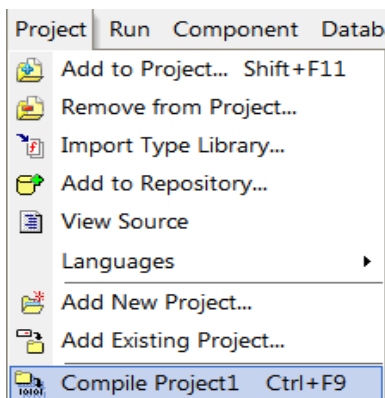


Рис. 2.12. Вибір команди **Compile Project** (Компілювати проєкт) з «*Головного*» меню **Project** (Проект)

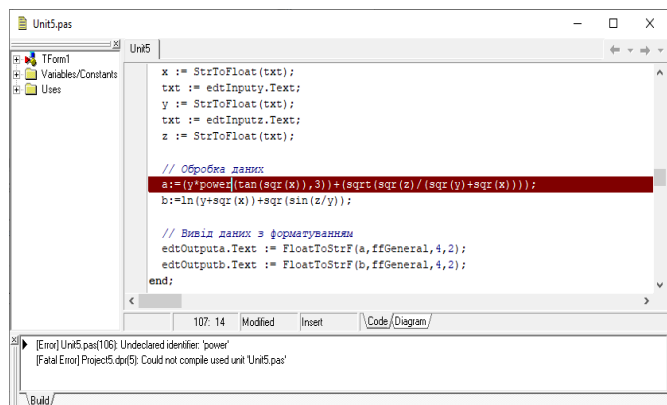


Рис. 2.13. Повідомлення компілятора про виявлені помилки

Якщо у програмі немає синтаксичних помилок, компілятор створює виконуваний файл програми. Ім'я виконуваного файлу таке ж, як і у файлу проекту, а розширення – exe. Delphi поміщає виконуваний файл у той самий каталог, де знаходиться файл проекту.

### 2.4.5. Запуск програми


Запуск програми може виконуватися з «*Головного*» меню **Run** (Запуск) обравши команду **Run** (Запуск) або натиснути на кнопці  **Run** (Запуск) на панелі інструментів «*Debug*» (Налагодження) (рис. 2.14) або натиснути клавішу **F9**.



Рис. 2.14. Панель інструментів «*Debug*» (Налагодження)

### 2.4.6. Помилки часу виконання

Під час роботи програми можуть виникати помилки, які називаються **помилками часу виконання** (run-time errors) або **винятками** (exceptions). У більшості випадків причинами **винятків** є невірні вихідні дані.

Після виникнення таких помилок програма перестане робити.

Необхідно перервати неробочу програму, вибравши із «*Головного*» меню **Run** (Запуск) команду **Program Reset** (Скидання програми) (рис. 2.15).

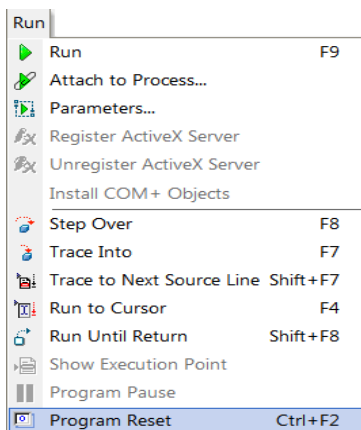


Рис.2.15. Вибір команди **Program Reset** (Скидання програми) із «*Головного*» меню **Run** (Запуск)

### 2.4.7. Захист полів вводу

Для виключення помилок, при вводі даних у поля вводу необхідно передбачити у вихідному коді програми захист полів вводу.

У лабораторній роботі №5 передбачені такі види захисту полів вводу (див. п. п. 2.4.3):

- захист полів вводу від вводу порожнього рядка;
- захист полів вводу  $x=$  та  $y=$  від вводу цифри «0» нуль;
- захист полів вводу від вводу літер та деяких символів, використовуючи подію OnKeyPress.

### 3. Зміст звіту з лабораторної роботи №5

**Звіт** з лабораторної роботи №5 складається з:

- титульного аркуша (див. Додаток Д);
- аркуша (див. Додаток Д) на якому розташовано: **тема** лабораторної роботи, **мета** лабораторної роботи, **розробка програми** (наведіть умову та формулу, які розглядаються в лабораторній роботі №5, див. п. п. 2.4);
- аркуша (див. Додаток Д) на якому розташовано: **створення інтерфейсу програми** (навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №5, див. п. п. 2.4.1);
- аркушів (див. Додаток Д) на яких розташовані: **лістинг програми** (навести **лістинг** програми, який розглядається в лабораторній роботі №5, див. п. п. 2.4.3) та **висновки**.

## Лабораторна робота №6

**Тема:** Діалоговий ввід-вивід даних.

**Мета роботи:** Засвоєння початкових знань та придбання навичок, необхідних для розробки програмних продуктів комп'ютерного забезпечення в технологічних процесах обробки металів різанням.

### Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Виконати алгоритм виконання лабораторної роботи №6.
3. Скласти звіт з лабораторної роботи №6 за наведеною формою (див. Додаток Є).

#### Хід виконання роботи

##### 1. Теоретичні відомості

Вивчити теоретичні відомості зі списку наведеної літератури.

##### 2. Алгоритм виконання лабораторної роботи №6

###### 2.1. Запуск програми Delphi

У системному меню «*Пуск*», операційної системи Windows вибрати:  
**Пуск > Програми > Borland Delphi 7 > Delphi 7.**

###### 2.2. Створення проєкту

Після запуску програми Delphi створюється за замовчуванням проєкт типу – **Application** (Програма). Delphi дозволяє розробляти різні програми, які в Delphi називаються проєктами. У лабораторних роботах створюються проєкти – **Application** (Програма), під операційну систему Windows з графічним інтерфейсом.

Для створення нового проєкту виберіть в «*Головному*» меню **File** (Файл) **New > Application** (Нова > Програма) (рис. 2.1).

### 2.3. Збереження проєкту

Delphi для кожного проєкту створює декілька файлів. Щоб файли різних проєктів не переплуталися між собою, слід для кожного проєкту створювати свою **папку** з іменем, яке містить назву та номер лабораторної роботи (наприклад, лабораторна робота 6 – Lab\_6).

Рекомендується імені проєкту надавати ім'я, яке містить номер лабораторної роботи (наприклад, проєкт для лабораторної роботи 6 – Project\_6).

**Файл проєкту** Project\_6 потрібно зберегти в папці Lab\_6.

Необхідно також зберегти **файл модуля вихідного коду**.

Рекомендується імені файлу модуля вихідного коду надавати ім'я, як у проєкті (наприклад, Unit\_6). Його також потрібно зберегти у папці Lab\_6.

На панелі інструментів «*Standard*» (Стандартна) натиснути кнопку **Save all** (Зберегти все) та зберегти **файли проєкту** та **файл модуля вихідного коду** згідно з вище сказаними рекомендаціями (рис. 2.2).

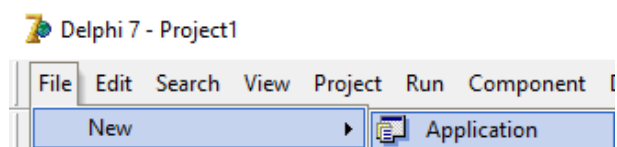


Рис. 2.1. Виберіть **New > Application** (Нова > Програма) в «*Головному*» меню **File** (Файл)

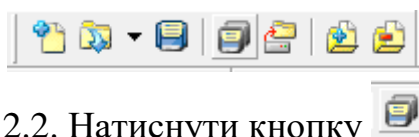


Рис. 2.2. Натиснути кнопку **Save all** (Зберегти все) на панелі інструментів «*Standard*» (Стандартна)

### 2.4. Розробка програми

Розробити програму обчислення значень змінних  $a$  та  $b$ , якщо:  $x = 2,1$ ;  $y = 0,59$  та  $z = -4,8$ , використовуючи формули:

$$a = y \cdot \operatorname{tg}^3(x^2) + \sqrt{\frac{z^2}{y^2 + x^2}}$$

$$b = \ln(y + x^2) + \sin^2\left(\frac{z}{x}\right)$$

У програмі передбачити ввід та вивід даних, а також вивід повідомлень у стандартні діалогові вікна.

## 2.4.1. Створення інтерфейсу програми

Інтерфейс програми представлений на рисунку 2.3. Ввід даних у стандартні діалогові вікна програми представлено на рисунку 2.4 (а, б, в). Вивід даних у стандартні діалогові вікна програми представлено на рисунку 2.5 (а, б). Вивід повідомлень у стандартні діалогові вікна програми представлено на рисунку 2.6 (а, б, в, г).

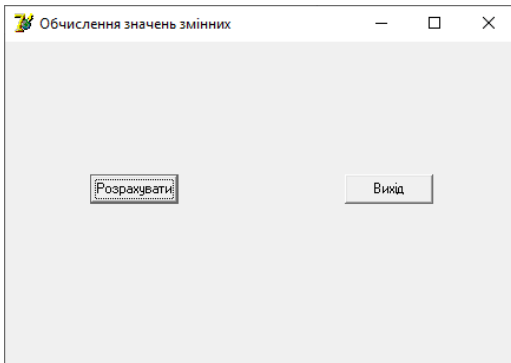


Рис. 2.3. Інтерфейс програми

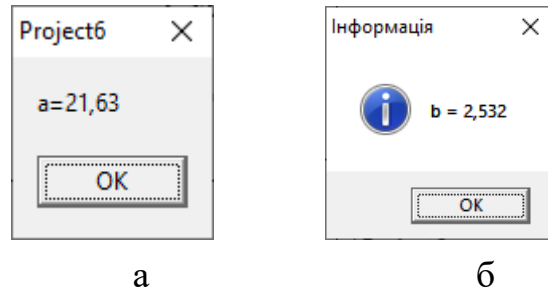
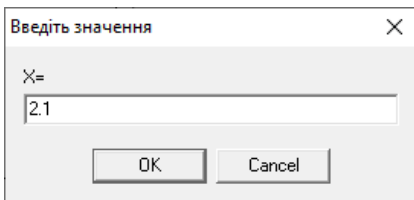
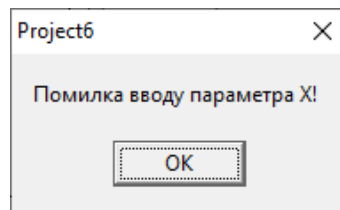


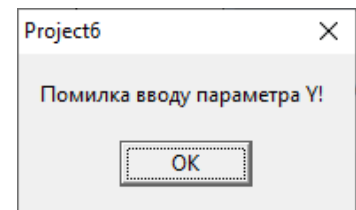
Рис. 2.5. Вивід даних у стандартні діалогові вікна програми: а, б



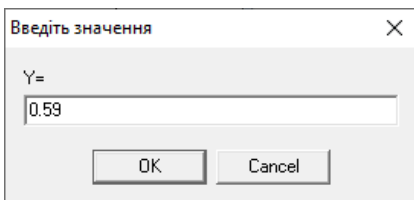
а



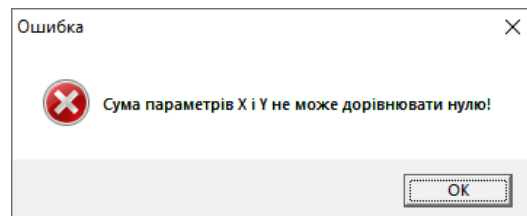
а



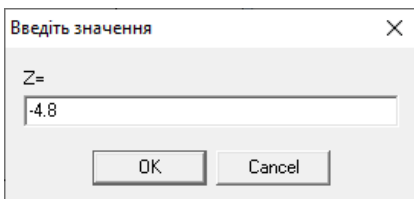
б



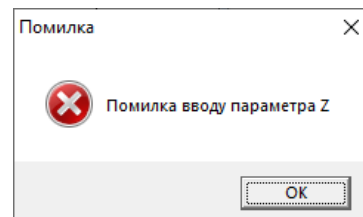
б



в



в



г

Рис. 2.4. Ввід даних у стандартні діалогові вікна програми: а, б, в

Рис. 2.6. Вивід повідомлень у стандартні діалогові вікна програми: а, б, в, г

### 2.4.1.1. Вікно форми Form1 (Форма1)

При створенні проекту типу – **Application** (Програма), Delphi автоматично створює вікно форми «**Form1**» (Форма1) (рис. 2.7). Вікно форми «**Form1**» (Форма1) являє собою шаблон вікна розробляємої програми.

### 2.4.1.2. Розміщення компонентів на вікні форми Form1 (Форма1)

На вікні форми «**Form1**» (Форма1) розмістити компоненти (рис. 2.8) вказані в таблиці 2.1.

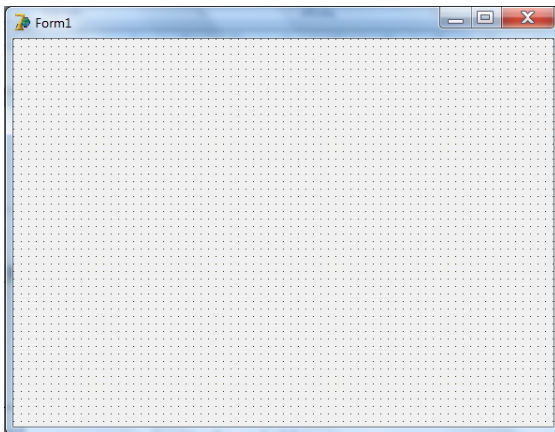


Рис. 2.7. Вікно форми «**Form1**» (Форма1)

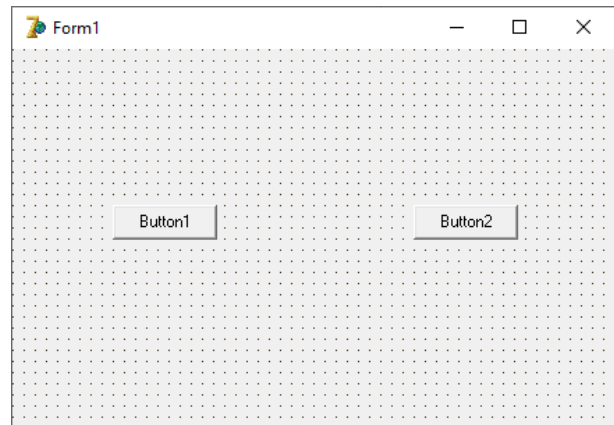


Рис. 2.8. Компоненти розміщені на вікні форми «**Form1**» (Форма1)

Для цього необхідно виконати наступні дії:

а) на «**Панелі компонентів**» на вкладці «**Standard**» (Стандартна) (рис. 2.9), вказати курсором миші на піктограму потрібного компонента (табл. 2.1);



б) потім вказати курсором миші у довільному місці на формі «**Form1**» (Форма1) (рис. 2.8);

в) якщо необхідно, перетягніть компонент у потрібне місце на формі «**Form1**» (Форма1) (рис. 2.8).



Рис. 2.9. «**Панель компонентів**» на вкладці «**Standard**» (Стандартна)

Таблиця 2.1 – Компоненти та їх властивості для інтерфейсу програми

Компонент	Піктограма	Властивість	Значення
Form1	-	Caption	Обчислення значень змінних
		BorderStyle	bsSingle
Button1		Name	btnCalculate
		Caption	Розрахувати
Button2		Name	btnExit
		Caption	Вихід

### 2.4.1.3. Зміна властивостей вікна форми Form1 (Форма1) та компонентів

Змінимо властивості вікна форми «*Form1*» (Форма1) та компонентів (рис. 2.10) згідно з вказаними властивостями в таблиці 2.1.

Для цього необхідно виконати наступні дії:

- а) у верхній частині вікна «*Object Inspector*» (Інспектор об'єктів) (рис. 2.11) вибрати ім'я об'єкта (форми або компонентів) (табл. 2.1);
- б) у лівій колонці вкладки «*Properties*» (Властивості) (рис. 2.11), де перелічені властивості об'єкта (форми чи компонентів) виділити властивість (табл. 2.1);
- в) у правій колонці вкладки «*Properties*» (Властивості) (рис. 2.11), де вказано значення об'єкта (форми або компонентів) вибрати або ввести значення властивості (табл. 2.1).

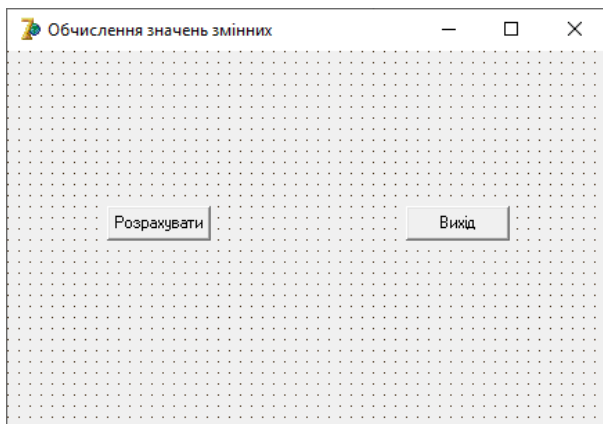


Рис. 2.10. Компоненти та вікно форми «*Form1*» (Форма1) із зміненими властивостями

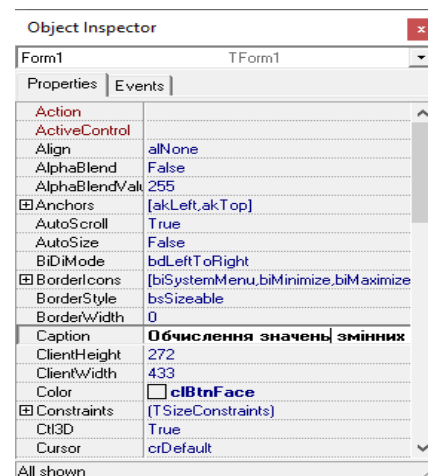


Рис. 2.11. Зміна значення властивості **Caption** (Заголовок) вікна форми «*Form1*» (Форма1) у вікні «*Object Inspector*» (Інспектор об'єктів) на **Обчислення значень змінних**

## 2.4.2. Створення обробників подій

Створимо обробники подій для компонентів, вказаних у таблиці 2.2.

Таблиця 2.2 – Обробники подій компонентів для програми

Компонент	Подія	Ім'я обробника
btnCalculate	OnClick	btnCalculateClick
btnExit	OnClick	btnExitClick

Для цього необхідно виконати наступні дії:

- а) у верхній частині вікна «*Object Inspector*» (Інспектор об'єктів) (рис. 2.12) вибрати ім'я об'єкта (форми або компонента) (табл. 2.2);
- б) у цьому ж вікні потрібно вибрати вкладку «*Events*» (Події) (рис. 2.12);
- в) у лівій колонці вкладки «*Events*» (Події) (рис. 2.12), де перелічені імена подій, які може використовувати вибраний об'єкт (компонент), вибрати ім'я події (табл. 2.2);
- г) для створення функції обробки події потрібно у правій колонці вкладки «*Events*» (Події) навпроти обраної раніше події зробити подвійне клацання мишею у полі імені обробника (рис. 2.12);
- д) відкриється вікно «*Unit\_6.pas*» (Редактор коду), до якого буде додано шаблон процедури обробника **btnExitClick** події **OnClick** (рис. 2.13), а у вікні «*Object Inspector*» (Інспектор об'єктів) поряд з ім'ям події з'явиться ім'я обробника (рис. 2.12);
- е) у вікні «*Unit\_6.pas*» (Редактор коду) між словами **begin** і **end** необхідно ввести інструкції, що реалізують обробку події (рис. 2.13). У лістингу програми, див. п. п. 2.4.3, наведені тексти інструкцій обробки події **OnClick** для кнопок **Розрахувати** (btnCalculate) та **Вихід** (btnExit).

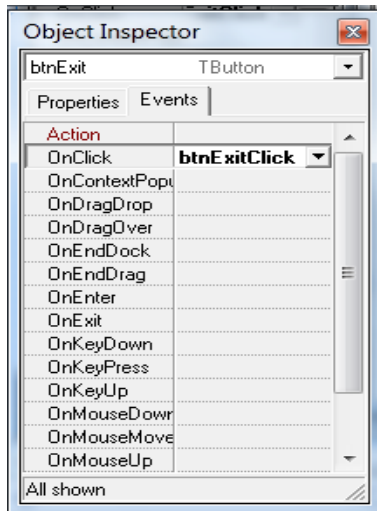


Рис. 2.12. Створення обробника **btnExitClick** події **OnClick** для компонента **btnExit**, на вкладці «*Events*» (Події) у вікні «*Object Inspector*» (Інспектор об'єктів)

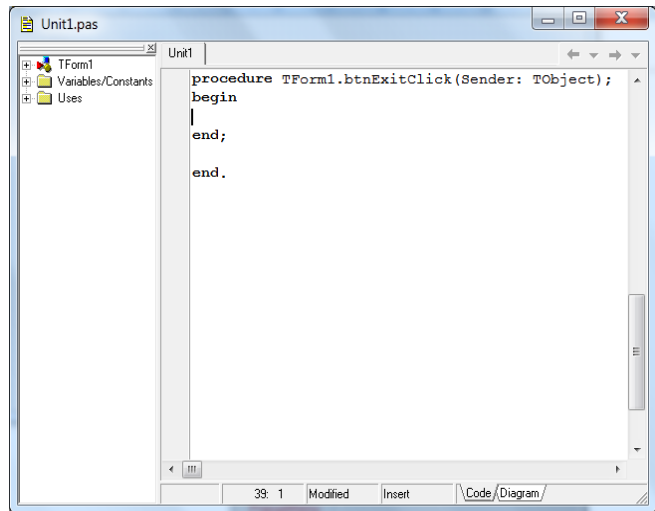


Рис. 2.13. Шаблон процедури обробника **btnExitClick** події **OnClick**, згенерований програмою Delphi

### 2.4.3. Створення вихідного коду

Вихідний код програми представлений у вигляді лістингу програми.

Так як у вихідному коді використовуються математичні функції, то необхідно в розділі **uses** додати модуль **Math**.

#### Лістинг програми

```

unit Unit_6;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Math;
type
  TForm1 = class(TForm)
    btnCalculate: TButton;
    btnExit: TButton;
  procedure btnCalculateClick(Sender: TObject);
  procedure btnExitClick(Sender: TObject);

```

```

private
{ Private declarations }
public
{ Public declarations }
end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.btnCalculateClick(Sender: TObject);
var
  // Оголошення змінних
  a, b, x, y, z : Real;
  code : Integer;
begin
  // Ввід даних з захистом полів вводу
  // X
  Val(InputBox('Введіть значення', 'X=', ' '), x, code);
  if (code <> 0) or (x = 0) then
  begin
  ShowMessage('Помилка вводу параметра X!');
  exit;
  end;
  // Y
  Val(InputBox('Введіть значення', 'Y=', ' '), y, code);
  if (code <> 0) or (y = 0) then
  begin
  ShowMessage('Помилка вводу параметра Y!');
  exit;
  end;
  if ((x+y)=0) then
  begin
  Application.MessageBox('Сума параметрів X і Y не може дорівнювати нулю!',
  'Ошибка', MB_OK+MB_ICONSTOP);

```

```

exit;
end;
// Z
Val(TextBox('Введіть значення', 'Z=', ' '), z, code);
if (code <> 0) or (z = 0) then
begin
Application.MessageBox('Помилка вводу параметра Z', 'Помилка',
MB_OK+MB_ICONSTOP);
exit;
end;
// Обробка даних
a:=(y*power(Tan(Sqr(x)),3))+(Sqrt(Sqr(z)/(Sqr(y)+Sqr(x))));
b:=ln(y+sqr(x))+sqr(sin(z/y));
// Вивід даних з форматуванням
ShowMessage('a='+FloatToStrF(a,ffGeneral,4,2));
Application.MessageBox(PChar(' b = '+FloatToStrF(b,ffGeneral,4,2)),
'Інформація', MB_OK+ MB_ICONINFORMATION);
end;
procedure TForm1.BtnExitClick(Sender: TObject);
begin
// Закриття програми
Application.Terminate;
end;
end.

```

#### 2.4.4. Компіляція програми

Процес компіляції складається із двох етапів. На першому етапі виконується перевірка тексту програми на відсутність синтаксичних помилок, а на другому – генерується виконувана програма (exe-файл).

**Компіляція програми** виконується з *«Головного»* меню **Project** (Проект) обравши команду **Compile Project** (Компілювати проект) (рис. 2.14) або натиснути клавіші **Ctrl+F9**.

Повідомлення про помилки, попередження та підказки відображаються в нижній частині вікна редактора коду (рис. 2.15).

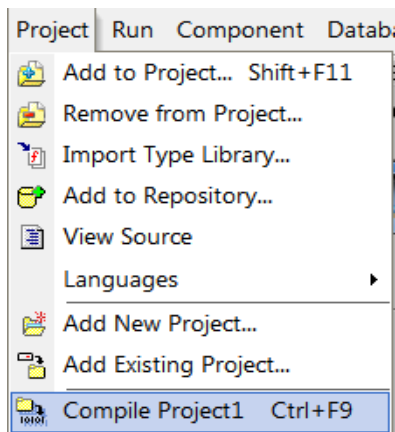


Рис. 2.14. Вибір команди **Compile Project** (Компілювати проєкт) з «Головного» меню **Project** (Проєкт)

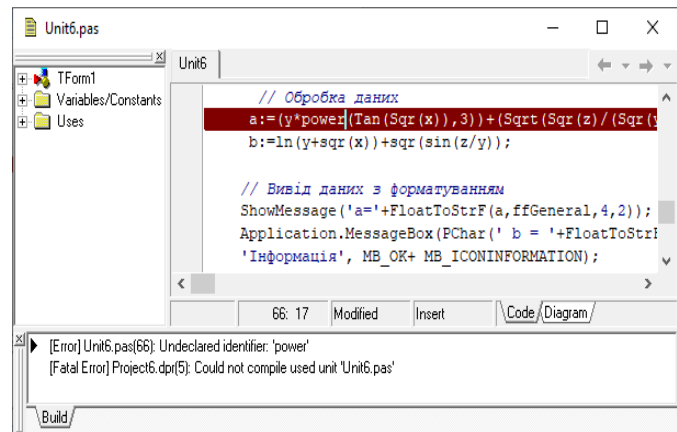


Рис. 2.15. Повідомлення компілятора про виявлені помилки

Процес усунення помилок має ітераційний характер. Зазвичай, спочатку усуваються найбільш очевидні помилки, наприклад, декларуються неоголошені змінні. Після чергового внесення змін до тексту програми виконується повторна компіляція. Слід зважати на те, що компілятор не завжди може точно локалізувати помилку. Тому аналізуючи фрагмент програми який на думку компілятора містить помилку, потрібно звертати увагу не тільки на той фрагмент коду, на який компілятор встановив курсор, але й на той, який знаходиться у попередньому рядку.

Якщо у програмі немає синтаксичних помилок, компілятор створює виконуваний файл програми. Ім'я виконуваного файлу таке ж, як і у файлу проєкту, а розширення – exe. Delphi поміщає виконуваний файл у той самий каталог, де знаходиться файл проєкту.

### 2.4.5. Запуск програми


Запуск програми може виконуватися з «Головного» меню **Run** (Запуск) обравши **Run** (Запуск) або натиснути на кнопці  **Run** (Запуск) на панелі інструментів «*Debug*» (Налагодження) (рис. 2.16) або натиснути клавішу **F9**.



Рис. 2.16. Панель інструментів «*Debug*» (Налагодження)

## 2.4.6. Помилки часу виконання

Під час роботи програми можуть виникати помилки, які називаються **помилками часу виконання** (run-time errors) або **винятками** (exceptions). У більшості випадків причинами **винятків** є невірні вихідні дані.

Після виникнення таких помилок програма перестане робити.

Необхідно перервати неробочу програму, вибравши із «**Головного**» меню **Run** (Запуск) команду **Program Reset** (Скидання програми) (рис. 2.17).

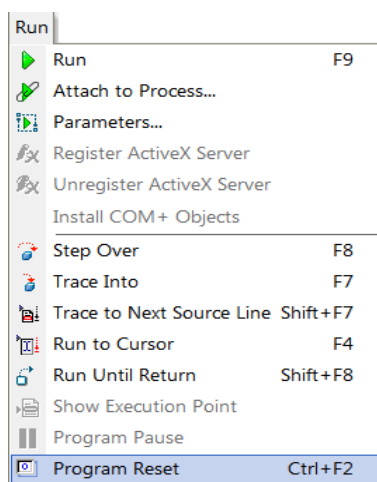


Рис. 2.17. Вибір команди **Program Reset** (Скидання програми) із «**Головного**» меню **Run** (Запуск)

## 2.4.7. Захист полів вводу

Для виключення помилок, при вводі даних у поля вводу необхідно передбачити у вихідному коді програми захист полів вводу.

У лабораторній роботі №6 використовуємо строкову функцію Val для захисту полів вводу (див. п. п. 2.4.3):

- від вводу порожнього рядка;
- від вводу цифри «0» нуль;
- від вводу літер та деяких символів.

## 3. Зміст звіту з лабораторної роботи №6

**Звіт** з лабораторної роботи №6 складається з:

- титульного аркуша (див. Додаток Є);

- аркуша (див. Додаток Є) на якому розташовано: **тема** лабораторної роботи, **мета** лабораторної роботи, **розробка програми** (наведіть умову та формулу, які розглядаються в лабораторній роботі №6, див. п. п. 2.4);

- аркуша (див. Додаток Є) на якому розташовано: **створення інтерфейсу програми** (навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №6, див. п. п. 2.4.1);

- аркушів (див. Додаток Є) на яких розташовані: **лістинг програми** (навести **лістинг** програми, який розглядається в лабораторній роботі №6, див. п. п. 2.4.3) та **висновки**.

### Запитання з самоконтролю до розділу 2

1. Що таке оператор?
2. Що таке операнд?
3. Для чого необхідно підключити модуль Math в операторі uses?
4. Для чого призначені математичні функції?
5. Для чого призначені строкові функції?
6. Якими компонентами можна скористатися для вводу числових значень?
7. Якими функціями можна скористатися при інтерактивному виводі даних?

## РОЗДІЛ 3. МЕТОДИ ВИБОРУ ДАНИХ

### Лабораторна робота №7

**Тема:** Структури прийняття рішень. Оператор if.

**Мета роботи:** Засвоєння початкових знань та придбання навичок, необхідних для розробки програмних продуктів комп'ютерного забезпечення в технологічних процесах обробки металів різанням.

### Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Виконати алгоритм виконання лабораторної роботи №7.
3. Скласти звіт з лабораторної роботи №7 за наведеною формою (див. Додаток Ж).

#### Хід виконання роботи

##### 1. Теоретичні відомості

Вивчити теоретичні відомості зі списку наведеної літератури.

##### 2. Алгоритм виконання лабораторної роботи №7

###### 2.1. Запуск програми Delphi

У системному меню «*Пуск*», операційної системи Windows, вибрати: **Пуск > Програми > Borland Delphi 7 > Delphi 7.**

###### 2.2. Створення проєкту

Після запуску програми Delphi створюється за замовчуванням проєкт типу – **Application** (Програма).

Delphi дозволяє розробляти різні програми, які в Delphi називаються проєктами. У лабораторних роботах створюються проєкти – **Application** (Програма), під операційну систему Windows з графічним інтерфейсом.

Для створення іншого проєкту, виберіть в «*Головному*» меню **File** (Файл) **New > Application** (Нова > Програма) (рис. 2.1).

### 2.3. Збереження проєкту

Delphi для кожного проєкту створює декілька файлів. Щоб файли різних проєктів не переплуталися між собою, слід для кожного проєкту створювати свою **папку** з іменем, яке містить назву та номер лабораторної роботи (наприклад, лабораторна робота 7 – Lab\_7).

Рекомендується імені проєкту надавати ім'я, яке містить номер лабораторної роботи (наприклад, проєкт для лабораторної роботи 7 – Project\_7).

**Файл проєкту** Project\_7 потрібно зберегти в папці Lab\_7.

Необхідно також зберегти **файл модуля вихідного коду**.

Рекомендується імені файлу модуля вихідного коду надавати ім'я, як у проєкті (наприклад, Unit\_7). Його також потрібно зберегти у папці Lab\_7.

На панелі інструментів «*Standard*» (Стандартна) натиснути кнопку **Save all** (Зберегти все) та зберегти **файли проєкту** та **файл модуля вихідного коду** згідно з вище сказаними рекомендаціями (рис. 2.2).

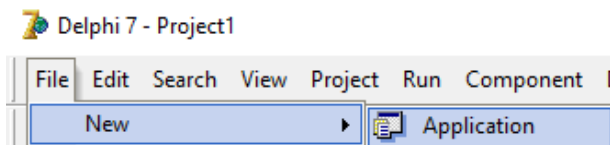


Рис. 2.1. Виберіть **New > Application** (Нова > Програма) в «*Головному*» меню **File** (Файл)

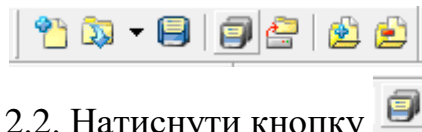


Рис. 2.2. Натиснути кнопку **Save all** (Зберегти все) на панелі інструментів «*Standard*» (Стандартна)

### 2.4. Розробка програми

Розробити програму для обчислення змінної  $y$ , якщо:

$x = -20 \dots -10$ , тоді  $y = x - 5$ ;

$x = 5 \dots 10$ , тоді  $y = \sqrt{x}$ .

Можливість вибору різних варіантів змінної  $x$  повинна бути врахована в інтерфейсі програми.

## 2.4.1. Створення інтерфейсу програми

Інтерфейс програми представлений на рисунку 2.3 (а, б). Вивід повідомлень програми представлено на рисунку 2.4 (а, б, в, г).

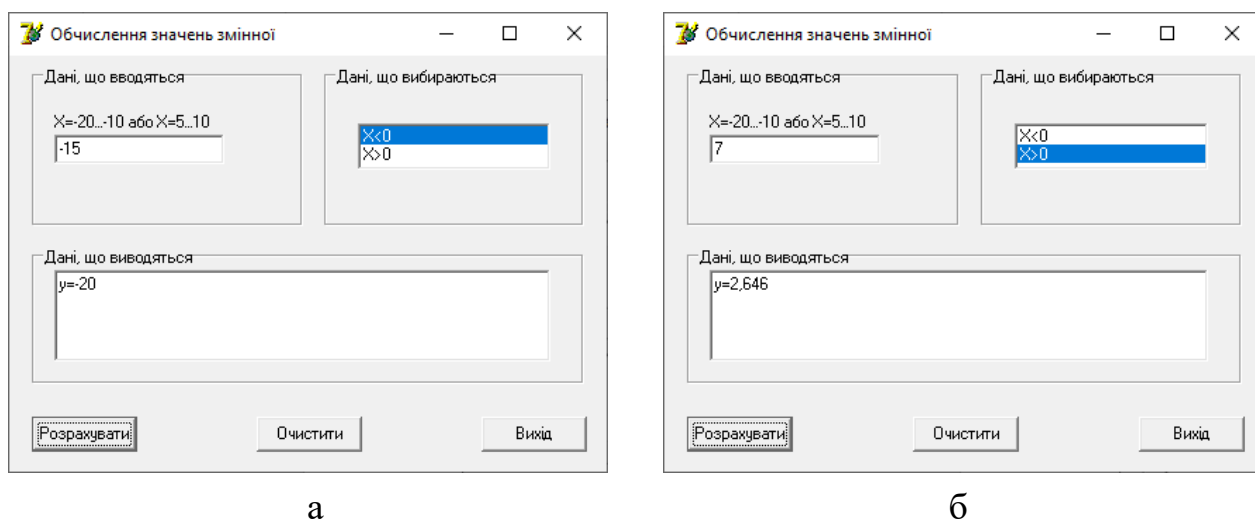


Рис. 2.3. Інтерфейс програми: а, б.

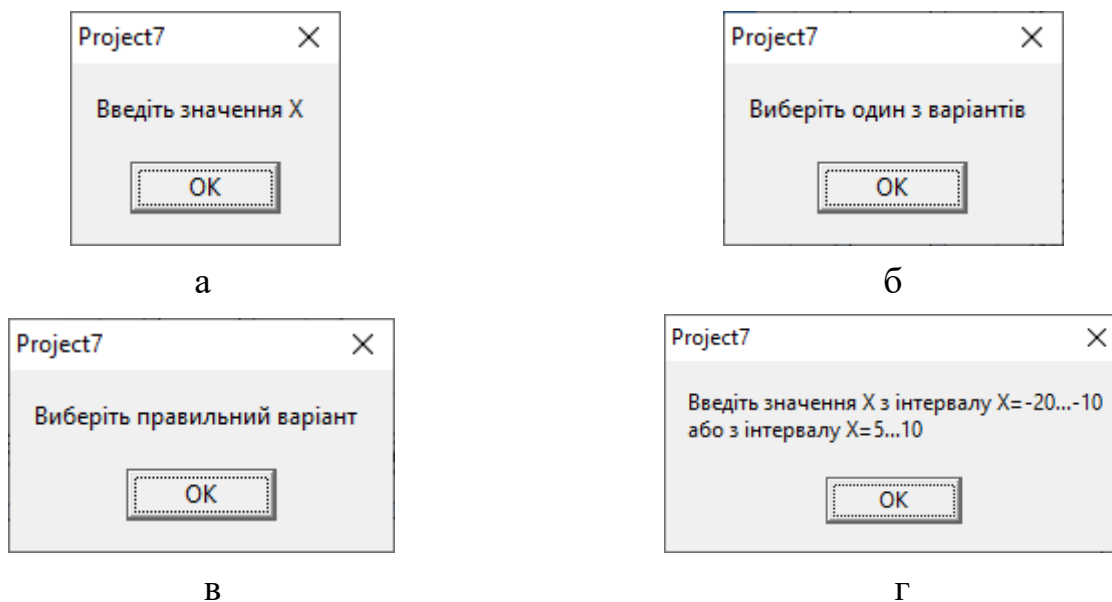


Рис. 2.4. Вивід повідомлень програми: а, б, в, г

### 2.4.1.1. Вікно форми Form1 (Форма1)

При створенні проекту типу – **Application** (Програма), Delphi автоматично створює вікно форми «**Form1**» (Форма1) (рис. 2.5). Вікно форми «**Form1**» (Форма1) являє собою шаблон вікна розробляємої програми.

## 2.4.1.2. Розміщення компонентів на вікні форми Form1 (Форма1)

На вікні форми «*Form1*» (Форма1) розмістити компоненти (рис. 2.6) вказані в таблиці 2.1.

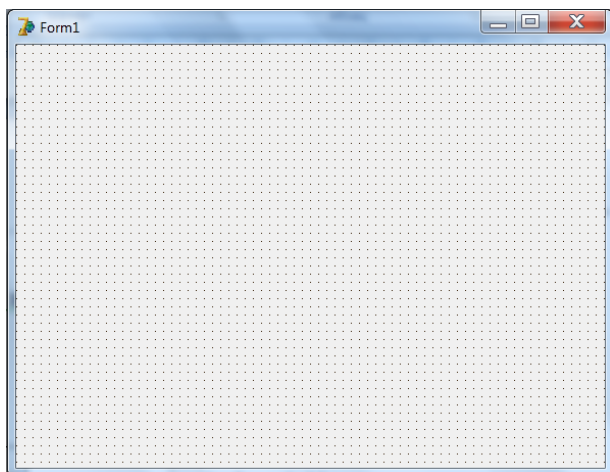


Рис. 2.5. Вікно форми «*Form1*» (Форма1)

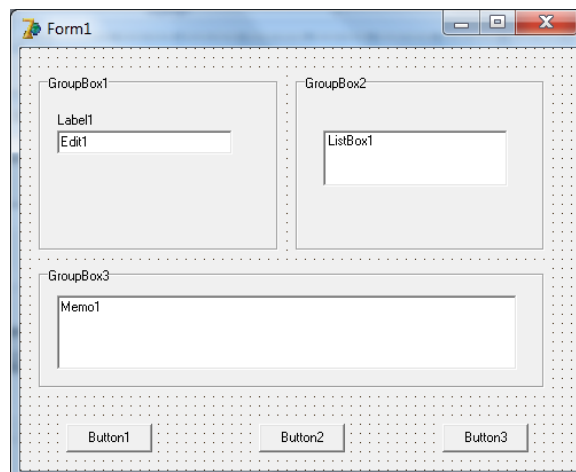


Рис. 2.6. Компоненти розміщені на вікні форми «*Form1*» (Форма1)

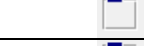






Для цього необхідно виконати наступні дії:

- а) на «*Панелі компонентів*» на вкладці «*Standard*» (Стандартна) (рис. 2.7) вказати курсором миші на піктограму потрібного компонента (табл. 2.1);
- б) потім вказати курсором миші у довільному місці на формі «*Form1*» (Форма1) (рис. 2.6);
- в) якщо необхідно, перетягніть компонент у потрібне місце на формі «*Form1*» (Форма1) (рис. 2.6).



Рис. 2.7. «*Панель компонентів*» на вкладці «*Standard*» (Стандартна)

Таблиця 2.1 – Компоненти та їх властивості для інтерфейсу програми

Компонент	Піктограма	Властивість	Значення
Form1		Caption	Обчислення значень змінної
		BorderStyle	bsSingle
Groupbox1		Caption	Дані, що вводяться
Groupbox2		Caption	Дані, що вибираються
Groupbox3		Caption	Дані, що виводяться
Edit1		Name	edtInputx
		Text	значення не вводимо, а тільки очистити поле
Label1		Caption	X=-20...-10 або X=5...10
Memo1		Name	memOutput
		Lines	натиснути кнопку  . У з'явившемся вікні очистити поле, потім натиснути кнопку <b>ОК</b>
		ReadOnly	True
ListBox		Name	lbChoice
		Items	натиснути кнопку  . У з'явившемся вікні вводимо: у першому рядку <b>X&lt;0</b> та натиснути <Enter>, а у другому рядку <b>X&gt;0</b> . Потім натиснути кнопку <b>ОК</b>
Button1		Name	btnCalculate
		Caption	Розрахувати
Button2		Name	btnClear
		Caption	Очистити
Button3		Name	btnExit
		Caption	Вихід

### 2.4.1.3. Зміна властивостей вікна форми Form1 (Форма1) та компонентів

Змінимо властивості вікна форми «*Form1*» (Форма1) та компонентів (рис. 2.8) згідно з вказаними властивостями в таблиці 2.1.

Для цього необхідно виконати наступні дії:

а) у верхній частині вікна «*Object Inspector*» (Інспектор об'єктів) (рис. 2.9) вибрати ім'я об'єкта (форми або компонентів) (табл. 2.1);

б) у лівій колонці вкладки «*Properties*» (Властивості) (рис. 2.9), де перелічені властивості об'єкта (форми чи компонентів) виділити властивість (табл. 2.1);

в) у правій колонці вкладки «*Properties*» (Властивості) (рис. 2.9), де вказано значення об'єкта (форми або компонентів) вибрати або ввести значення властивості (табл. 2.1).

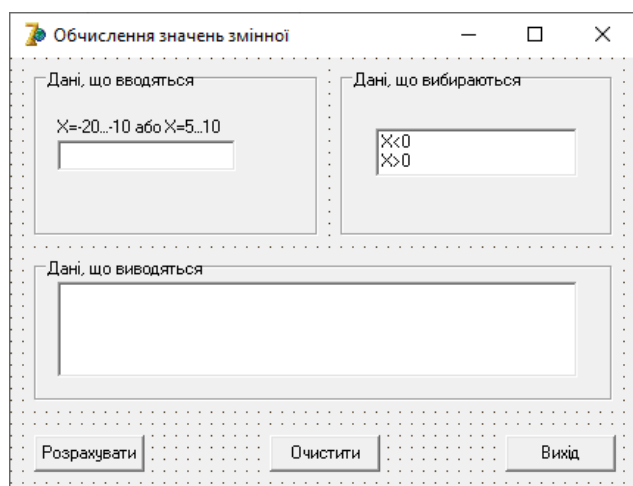


Рис. 2.8. Компоненти та вікно форми «*Form1*» (Форма1) із зміненими властивостями

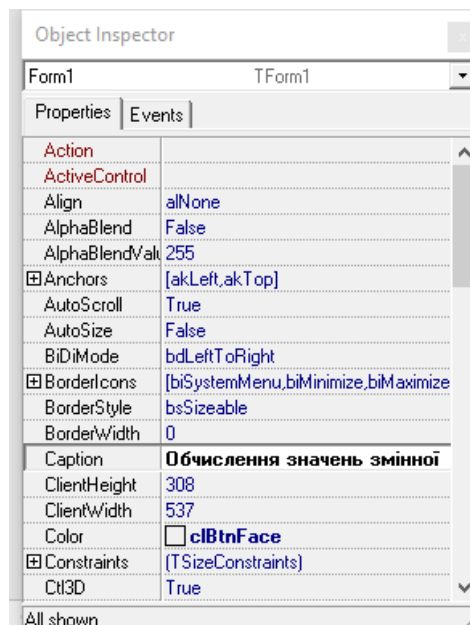


Рис. 2.9. Зміна значення властивості **Caption** (Заголовок) вікна форми «*Form1*» (Форма1) у вікні «*Object Inspector*» (Інспектор об'єктів) на **Обчислення значень змінної**

### 2.4.2. Створення обробників подій

Створимо обробники подій для компонентів, вказаних у таблиці 2.2.

Таблиця 2.2 – Обробники подій компонентів для програми

Компонент	Подія	Ім'я обробника
btnCalculate	OnClick	btnCalculateClick
btnClear	OnClick	btnClearClick
btnExit	OnClick	btnExitClick

Для цього необхідно виконати наступні дії:

- у верхній частині вікна «*Object Inspector*» (Інспектор об'єктів) (рис. 2.10) вибрати ім'я об'єкта (форми або компонента) (табл. 2.2);
- у цьому ж вікні потрібно вибрати вкладку «*Events*» (Події) (рис. 2.10);
- у лівій колонці вкладки «*Events*» (Події) (рис. 2.10), де перелічені імена

подій, які може використовувати вибраний об'єкт (компонент), вибрати ім'я події (табл. 2.2);

г) для створення функції обробки події потрібно у правій колонці вкладки «*Events*» (Події) навпроти обраної раніше події зробити подвійне клацання мишею у полі імені обробника (рис. 2.10);

д) відкриється вікно «*Unit\_7.pas*» (Редактор коду), до якого буде додано шаблон процедури обробника **btnExitClick** події **OnClick** (рис. 2.11), а у вікні «*Object Inspector*» (Інспектор об'єктів) поряд з ім'ям події з'явиться ім'я обробника (рис. 2.10);

е) у вікні «*Unit\_7.pas*» (Редактор коду) між словами **begin** і **end** необхідно ввести інструкції, що реалізують обробку події (рис. 2.11). У лістингу програми, див. п. п. 2.4.3, наведені тексти інструкцій обробки події **OnClick** для кнопок: **Розрахувати** (btnCalculate), **Очистити** (btnClear) та **Вихід** (btnExit).

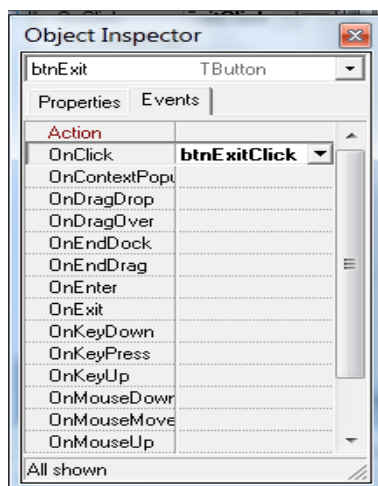


Рис. 2.10. Створення обробника **btnExitClick** події **OnClick** для компонента btnExit, на вкладці «*Events*» (Події) у вікні «*Object Inspector*» (Інспектор об'єктів)

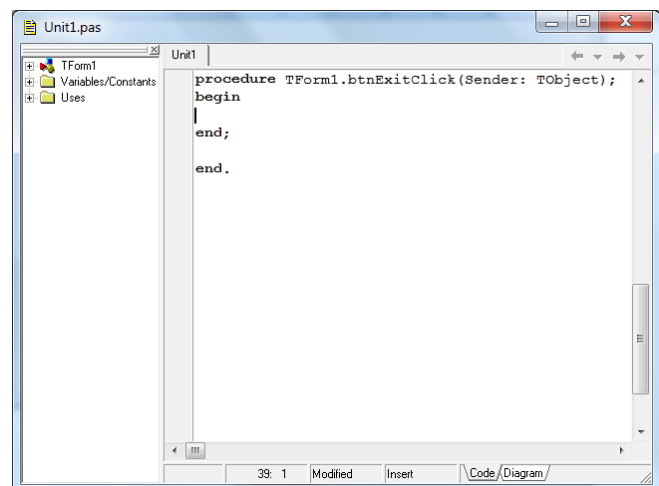


Рис. 2.11. Шаблон процедури обробника **btnExitClick** події **OnClick**, згенерований програмою Delphi

### 2.4.3. Створення вихідного коду

Вихідний код програми представлений у вигляді лістингу програми.

#### Лістинг програми

```
unit Unit_7;  
interface  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, ExtCtrls, StdCtrls;  
type  
  TForm1 = class(TForm)  
    edtinputx: TEdit;  
    btnCalculate: TButton;  
    btnClear: TButton;  
    btnExit: TButton;  
    Label1: TLabel;  
    GroupBox1: TGroupBox;  
    GroupBox2: TGroupBox;  
    GroupBox3: TGroupBox;  
    memOutput: TMemo;  
    lbChoice: TListBox;  
    procedure btnCalculateClick(Sender: TObject);  
    procedure btnClearClick(Sender: TObject);  
    procedure btnExitClick(Sender: TObject);  
  private  
    { Private declarations }  
  public  
    { Public declarations }  
end;  
var  
  Form1: TForm1;  
implementation
```

```

{$R *.dfm}
procedure TForm1.btnCalculateClick(Sender: TObject);
var
    // Оголошення змінних
    x, y : real;
    code : integer;
begin
    // Ввід даних із захистом поля вводу
    // x
    val(edinputx.text, x, code);
    if code<>0 then
        begin
            showmessage('Введіть значення X');
            edinputx.SetFocus;
            exit;
        end
    else
        if (x<-20) or ((x>-10) and (x<5)) or (x>10) then
            begin
                showmessage('Введіть значення X з інтервалу X=-20...-10'+#13#10+
                'або з інтервалу X=5...10');
                edinputx.SetFocus;
                exit;
            end;
        // Прийняття рішення та обробка даних
        if (lbChoice.ItemIndex=0) and (x<0) then y:=x-5
        else
            if (lbChoice.ItemIndex=1) and (x>0) then y:=sqrt(x)
            else
                if (lbChoice.ItemIndex=0) and (x>0) or
                (lbChoice.ItemIndex=1) and (x<0) then
                    begin
                        showmessage('Виберіть правильний варіант');
                        exit;
                    end

```

```

end
else
begin
showmessage('Виберіть один з варіантів');
exit;
end;
// Очищення полів виводу
memoutput.Lines.Clear;
// Вивід даних з форматуванням
memoutput.Lines.Add('y=' + FloatToStrF(y,ffGeneral,4,2));
end;
procedure TForm1.btnExitClick(Sender: TObject);
begin
// Закриття програми
Close;
end;
procedure TForm1.btnClearClick(Sender: TObject);
begin
// Очищення поля вводу
edtinputx.Text:=' ';
// Очищення поля вибору
lbChoice.Itemindex:=-1;
// Очищення поля виводу
memoutput.lines.Clear;
end;
end.

```

#### 2.4.4. Компіляція програми

Процес компіляції складається із двох етапів. На першому етапі виконується перевірка тексту програми на відсутність синтаксичних помилок, а на другому – генерується виконувана програма (exe-файл).

**Компіляція програми** виконується з «*Головного*» меню **Project** (Проект) обравши команду **Compile Project** (Компілювати проєкт) (рис. 2.12) або

натиснути клавіші **Ctrl+F9**.

Повідомлення про помилки, попередження та підказки відображаються в нижній частині вікна редактора коду (рис. 2.13).

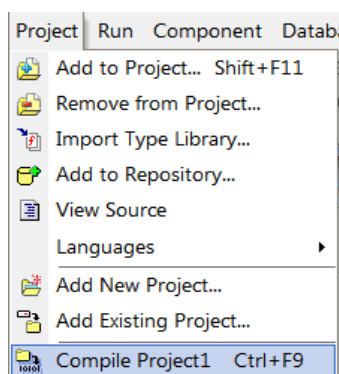


Рис. 2.12. Вибір команди **Compile Project** (Компілювати проєкт) з «Головного» меню **Project** (Проект)

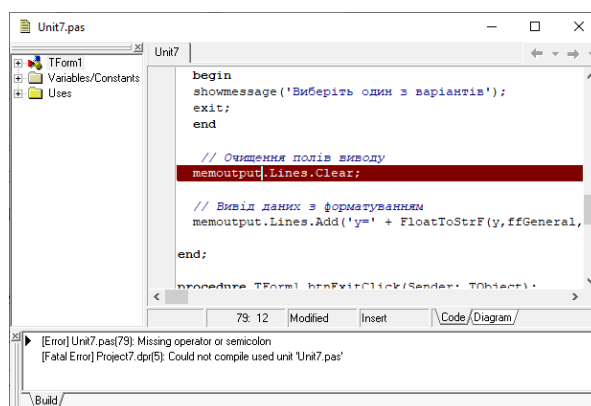



Рис. 2.13. Повідомлення компілятора про виявлені помилки

Процес усунення помилок має ітераційний характер. Зазвичай, спочатку усуваються найбільш очевидні помилки, наприклад, декларуються неоголошені змінні. Після чергового внесення змін до тексту програми виконується повторна компіляція. Слід зважати на те, що компілятор не завжди може точно локалізувати помилку. Тому аналізуючи фрагмент програми який на думку компілятора містить помилку, потрібно звертати увагу не тільки на той фрагмент коду, на який компілятор встановив курсор, але й на той, який знаходиться у попередньому рядку.

Якщо у програмі немає синтаксичних помилок, компілятор створює виконуваний файл програми. Ім'я виконуваного файлу таке ж, як і у файлу проєкту, а розширення – *exe*. Delphi поміщає виконуваний файл у той самий каталог, де знаходиться файл проєкту.

### 2.4.5. Запуск програми

Запуск програми може виконуватися з «Головного» меню **Run** (Запуск) обравши команду **Run** (Запуск) або натиснути на кнопці  **Run** (Запуск) на панелі інструментів «*Debug*» (Налагодження) (рис. 2.14) або натиснути клавішу **F9**.

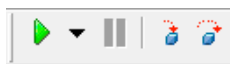


Рис. 2.14. Панель інструментів «*Debug*» (Налагодження)

### 2.4.6. Помилки часу виконання

Під час роботи програми можуть виникати помилки, які називаються **помилками часу виконання** (run-time errors) або **винятками** (exceptions). У більшості випадків причинами **винятків** є невірні вихідні дані.

Після виникнення таких помилок програма перестане робити.

Необхідно перервати неробочу програму, вибравши із «*Головного*» меню **Run** (Запуск) команду **Program Reset** (Скидання програми) (рис. 2.15).

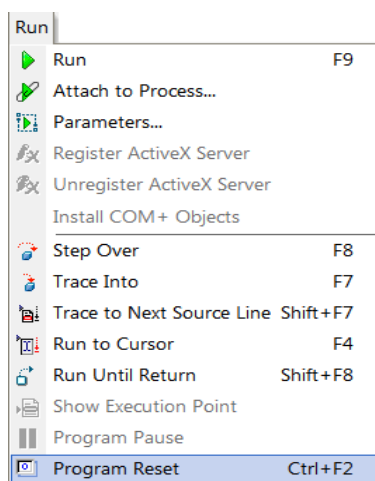


Рис. 2.15. Вибір команди **Program Reset** (Скидання програми) із «*Головного*» меню **Run** (Запуск)

### 2.4.7. Захист полів вводу

Для виключення помилок, при вводі даних у поля вводу необхідно передбачити у вихідному коді програми захист полів вводу.

У лабораторній роботі №7 використовуємо рядкову функцію Val, яка дозволяє використовувати такі види захисту поля вводу (див. п. п. 2.4.3):

- захист поля вводу від вводу порожнього рядка;
- захист поля вводу від вводу цифри «0» нуль;
- захист поля вводу від вводу літер та деяких символів.

### 3. Зміст звіту з лабораторної роботи №7

**Звіт** з лабораторної роботи №7 складається з:

- титульного аркуша (див. Додаток Ж);
- аркуша (див. Додаток Ж) на якому розташовано: **тема** лабораторної роботи, **мета** лабораторної роботи, **розробка програми** (наведіть умову та **формулу**, які розглядаються в лабораторній роботі №7, див. п. п. 2.4);
- аркуша (див. Додаток Ж) на якому розташовано: **створення інтерфейсу програми** (навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №7, див. п. п. 2.4.1);
- аркушів (див. Додаток Ж) на яких розташовані: **лістинг програми** (навести **лістинг** програми, який розглядається в лабораторній роботі №7, див. п. п. 2.4.3) та **висновки**.

## Лабораторна робота №8

**Тема:** Структури прийняття рішень. Оператор case.

**Мета роботи:** Засвоєння початкових знань та придбання навичок, необхідних для розробки програмних продуктів комп'ютерного забезпечення в технологічних процесах обробки металів різанням.

### Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Виконати алгоритм виконання лабораторної роботи №8.
3. Скласти звіт з лабораторної роботи №8 за наведеною формою (див. Додаток 3).

#### Хід виконання роботи

##### 1. Теоретичні відомості

Вивчити теоретичні відомості зі списку наведеної літератури.

##### 2. Алгоритм виконання лабораторної роботи №8

###### 2.1. Запуск програми Delphi

У системному меню «*Пуск*», операційної системи Windows, вибрати: **Пуск > Програми > Borland Delphi 7 > Delphi 7.**

###### 2.2. Створення проєкту

Після запуску програми Delphi створюється за замовчуванням проєкт типу – **Application** (Програма).

Delphi дозволяє розробляти різні програми, які в Delphi називаються проєктами. У лабораторних роботах створюються проєкти – **Application** (Програма), під операційну систему Windows з графічним інтерфейсом.

Для створення нового проєкту виберіть в «*Головному*» меню **File** (Файл) **New > Application** (Нова > Програма) (рис.2.1).

## 2.3. Збереження проєкту

Delphi для кожного проєкту створює декілька файлів. Щоб файли різних проєктів не переплуталися між собою, слід для кожного проєкту створювати свою **папку** з іменем, яке містить назву та номер лабораторної роботи (наприклад, лабораторна робота 8 – Lab\_8).

Рекомендується імені проєкту надавати ім'я, яке містить номер лабораторної роботи (наприклад, проєкт для лабораторної роботи 8 – Project\_8).

**Файл проєкту** Project\_8 потрібно зберегти в папці Lab\_8.

Необхідно також зберегти **файл модуля вихідного коду**.

Рекомендується імені файлу модуля вихідного коду надавати ім'я, як у проєкті (наприклад, Unit\_8). Його також потрібно зберегти у папці Lab\_8.

На панелі інструментів «*Standard*» (Стандартна) натиснути кнопку **Save all** (Зберегти все) та зберегти **файли проєкту** та **файл модуля вихідного коду** згідно з вище сказаними рекомендаціями (рис. 2.2).

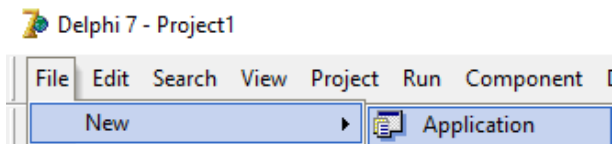


Рис. 2.1. Виберіть **New > Application** (Нова > Програма) в «*Головному*» меню **File** (Файл)

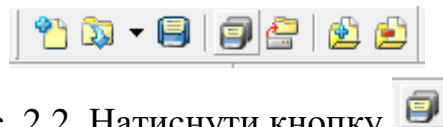


Рис. 2.2. Натиснути кнопку **Save all** (Зберегти все) на панелі інструментів **Standard** (Стандартна)

## 2.4. Розробка програми

Розробити програму у вигляді тесту на вибір компонентів латунного сплаву.

### 2.4.1. Створення інтерфейсу програми

Інтерфейс програми представлений на рисунку 2.3. Вивід повідомлень програми представлено на рисунку 2.4 (а, б, в, г).

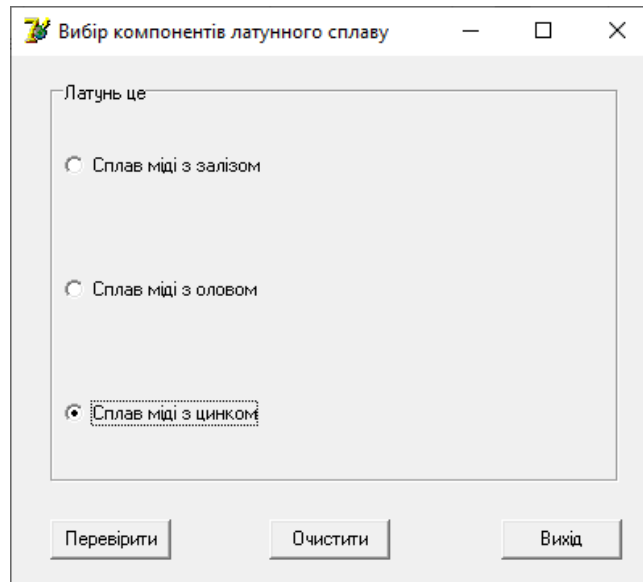


Рис. 2.3. Інтерфейс програми

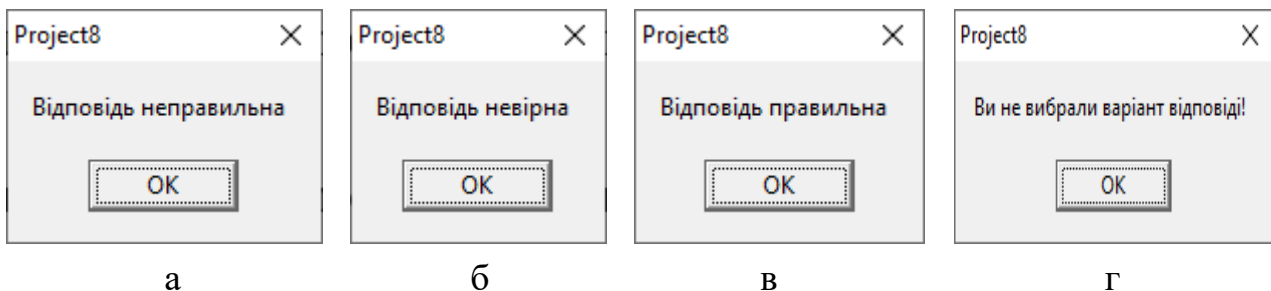


Рис. 2.4. Вивід повідомлень програми: а, б, в, г

#### 2.4.1.1. Вікно форми Form1 (Форма1)

При створенні проєкту типу – **Application** (Програма), Delphi автоматично створює вікно форми «**Form1**» (Форма1) (рис. 2.5). Вікно форми «**Form1**» (Форма1) являє собою шаблон вікна розробляємої програми.

#### 2.4.1.2. Розміщення компонентів на вікні форми Form1 (Форма1)

На вікні форми «**Form1**» (Форма1) розмістити компоненти (рис. 2.6) вказані в таблиці 2.1.

Для цього необхідно виконати наступні дії:

а) на «**Панелі компонентів**» на вкладці «**Standard**» (Стандартна) (рис. 2.7), вказати курсором миші на піктограму потрібного компонента (табл. 2.1);

б) потім вказати курсором миші у довільному місці на формі «*Form1*» (Форма1) (рис. 2.6);

в) якщо необхідно, перетягніть компонент у потрібне місце на формі «*Form1*» (Форма1) (рис. 2.6).

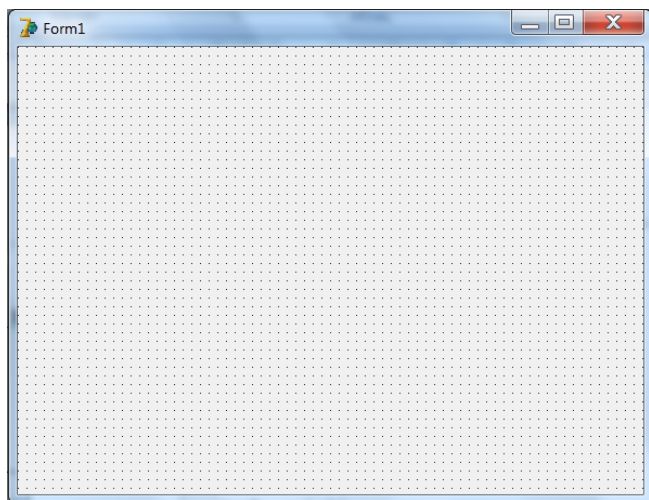


Рис. 2.5. Вікно форми «*Form1*» (Форма1)

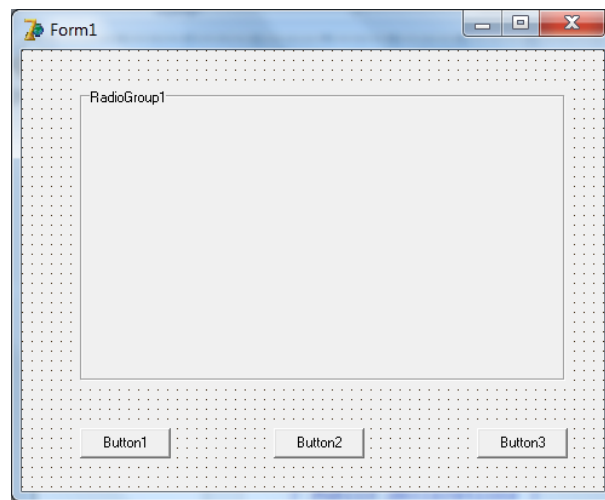


Рис. 2.6. Компоненти розміщені на вікні форми «*Form1*» (Форма1)

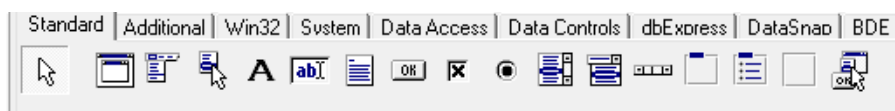





Рис. 2.7. «Панель компонентів» на вкладці «*Standard*» (Стандартна)

Таблиця 2.1 – Компоненти та їх властивості для інтерфейсу програми

Компонент	Піктограма	Властивість	Значення
Form1	-	Caption	Вибір компонентів латунного сплаву
		BorderStyle	bsSingle
RadioGroup1		Name	rgChoice
		Caption	Латунь це:
		Items	натиснути кнопку . У з'явившемуся вікні вводимо: у першому рядку <b>Сплав міді з залізом</b> та натиснути <Enter>; у другому рядку <b>Сплав міді з оловом</b> та натиснути <Enter>; у третьому рядку <b>Сплав міді з цинком</b> та натиснути кнопку <b>ОК</b>

## Продовження таблиці 2.1

Компонент	Піктограма	Властивість	Значення
Button1		Name	btnCheck
		Caption	Перевірити
Button2		Name	btnClear
		Caption	Очистити
Button3		Name	btnExit
		Caption	Вихід

### 2.4.1.3. Зміна властивостей вікна форми Form1 (Форма1) та компонентів

Змінимо властивості вікна форми «*Form1*» (Форма1) та компонентів (рис. 2.8) згідно з вказаними властивостями в таблиці 2.1.

Для цього необхідно виконати наступні дії:

а) у верхній частині вікна «*Object Inspector*» (Інспектор об'єктів) (рис. 2.9) вибрати ім'я об'єкта (форми або компонентів) (табл. 2.1);

б) у лівій колонці вкладки «*Properties*» (Властивості) (рис. 2.9), де перелічені властивості об'єкта (форми чи компонентів) виділити властивість (табл. 2.1);

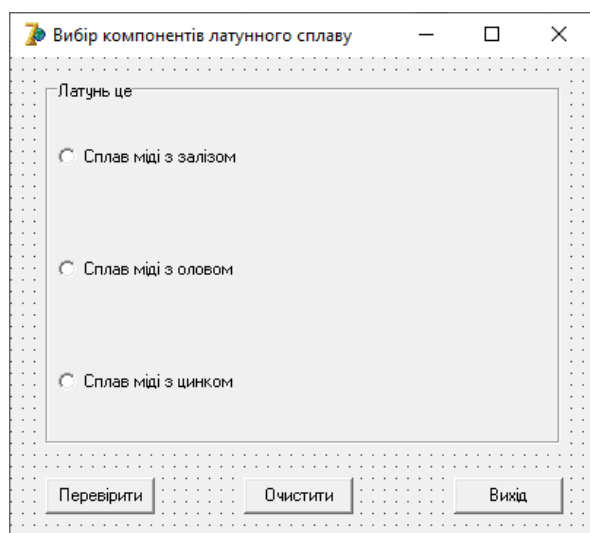


Рис. 2.8. Компоненти та вікно форми «*Form1*» (Форма1) із зміненими властивостями

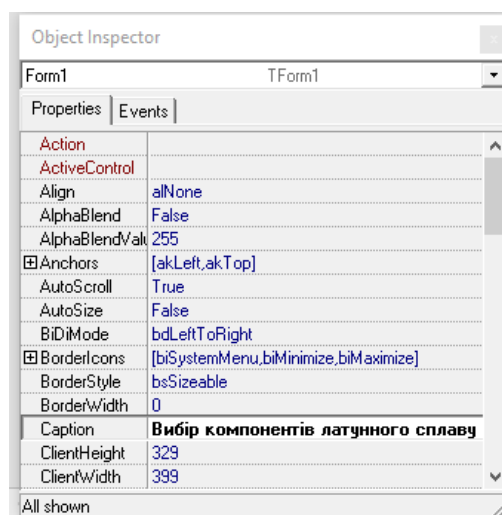


Рис. 2.9. Зміна значення властивості **Caption** (Заголовок) вікна форми «*Form1*» (Форма1) у вікні «*Object Inspector*» (Інспектор об'єктів) на **Вибір компонентів латунного сплаву**

в) у правій колонці вкладки «*Properties*» (Властивості) (рис. 2.9), де вказано значення об'єкта (форми або компонентів) вибрати або ввести значення властивості (табл. 2.1).

## 2.4.2. Створення обробників подій

Створимо обробники подій для компонентів, вказаних у таблиці 2.2.

Таблиця 2.2 – Обробники подій компонентів для програми

Компонент	Подія	Ім'я обробника
btnCheck	OnClick	btnCheckClick
btnClear	OnClick	btnClearClick
btnExit	OnClick	btnExitClick

Для цього необхідно виконати наступні дії:

- а) у верхній частині вікна «*Object Inspector*» (Інспектор об'єктів) (рис. 2.10) вибрати ім'я об'єкта (форми або компонента) (табл. 2.2);
- б) у цьому ж вікні потрібно вибрати вкладку «*Events*» (Події) (рис. 2.10);
- в) у лівій колонці вкладки «*Events*» (Події) (рис. 2.10), де перелічені імена подій, які може використовувати вибраний об'єкт (компонент), вибрати ім'я події (табл. 2.2);
- г) для створення функції обробки події потрібно у правій колонці вкладки «*Events*» (Події) навпроти обраної раніше події зробити подвійне клацання мишею у полі імені обробника (рис. 2.10);
- д) відкриється вікно «*Unit\_8.pas*» (Редактор коду), до якого буде додано шаблон процедури обробника **btnExitClick** події **OnClick** (рис. 2.11), а у вікні «*Object Inspector*» (Інспектор об'єктів) поряд з ім'ям події з'явиться ім'я обробника (рис. 2.10);
- е) у вікні «*Unit\_8.pas*» (Редактор коду) між словами **begin** і **end** необхідно ввести інструкції, що реалізують обробку події (рис. 2.11). У лістингу програми, див. п. п. 2.4.3, наведені тексти інструкцій: обробки події **OnClick** для кнопок: **Перевірити** (btnCheck), **Очистити** (btnClear) та **Вихід** (btnExit).

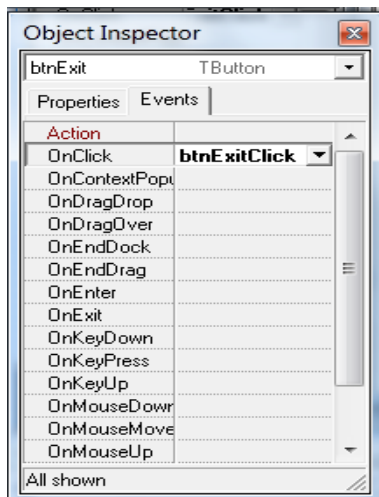


Рис. 2.10. Створення обробника **btnExitClick** події **OnClick** для компонента **btnExit**, на вкладці «*Events*» (Події) у вікні «*Object Inspector*» (Інспектор об'єктів)

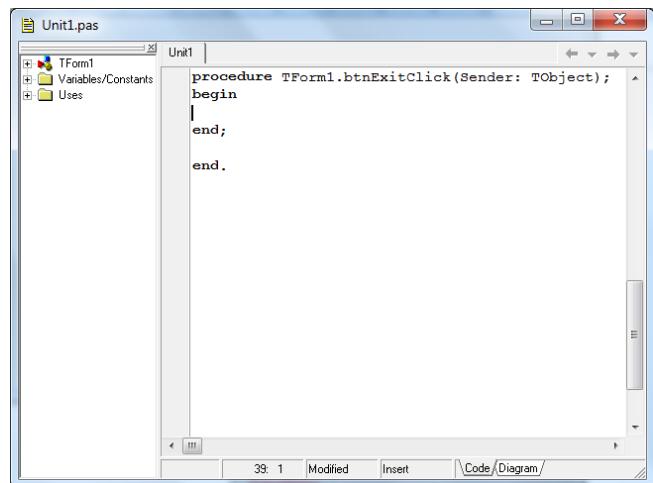


Рис. 2.11. Шаблон процедури обробника **btnExitClick** події **OnClick**, згенерований програмою Delphi

### 2.4.3. Створення вихідного коду

Вихідний код програми представлений у вигляді лістингу програми.

#### Лістинг програми

```

unit Unit_8;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls;
type
  TForm1 = class(TForm)
    btnCheck: TButton;
    btnClear: TButton;
    btnExit: TButton;
    rgChoice: TRadioGroup;
  procedure btnCheckClick(Sender: TObject);
  procedure btnClearClick(Sender: TObject);

```

```

procedure btnExitClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.btnCheckClick(Sender: TObject);
begin
  // Вибір варіанта відповіді
  case rgChoice.ItemIndex of
  // Виведення повідомлень під час вибору варіанта відповіді
  0: ShowMessage('Відповідь неправильна');
  1: ShowMessage('Відповідь невірна');
  2: ShowMessage('Відповідь правильна');
  else
  // Виведення повідомлення, якщо не вибрано варіант відповіді
  ShowMessage('Ви не вибрали варіант відповіді!')
  end;
end;
procedure TForm1.btnClearClick(Sender: TObject);
begin
  // Очищення полів вибору
  rgChoice.Itemindex:=-1;
end;
procedure TForm1.btnExitClick(Sender: TObject);
begin
  // Закриття програми
  Close;
end;
end.

```

## 2.4.4. Компіляція програми

Процес компіляції складається із двох етапів. На першому етапі виконується перевірка тексту програми на відсутність синтаксичних помилок, а на другому – генерується виконувана програма (exe-файл).

**Компіляція програми** виконується з «*Головного*» меню **Project** (Проект) обравши команду **Compile Project** (Компілювати проєкт) (рис. 2.12) або натиснути клавіші **Ctrl+F9**.

Повідомлення про помилки, попередження та підказки відображаються в нижній частині вікна редактора коду (рис. 2.13).

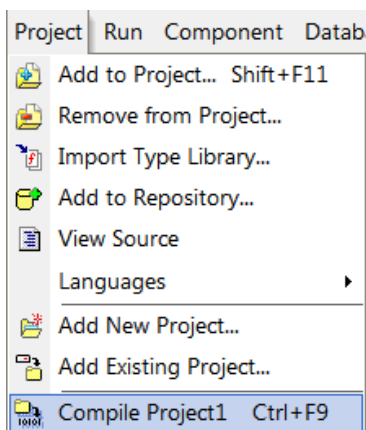


Рис. 2.12. Вибір команди **Compile Project** (Компілювати проєкт) з «*Головного*» меню **Project** (Проект)

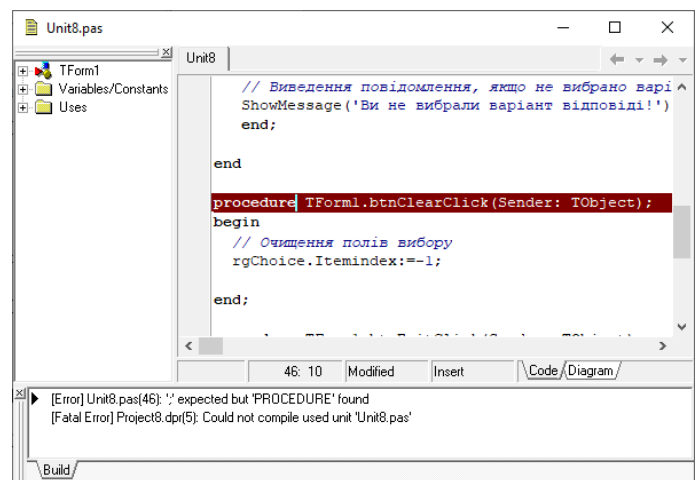


Рис. 2.13. Повідомлення компілятора про виявлені помилки

Процес усунення помилок має ітераційний характер. Зазвичай, спочатку усуваються найбільш очевидні помилки, наприклад, декларуються неоголошені змінні. Після чергового внесення змін до тексту програми виконується повторна компіляція. Слід зважати на те, що компілятор не завжди може точно локалізувати помилку. Тому аналізуючи фрагмент програми який на думку компілятора містить помилку, потрібно звертати увагу не тільки на той фрагмент коду, на який компілятор встановив курсор, але й на той, який знаходиться у попередньому рядку.

Якщо у програмі немає синтаксичних помилок, компілятор створює виконуваний файл програми. Ім'я виконуваного файлу таке ж, як і у файлу проєкту, а розширення – exe. Delphi поміщає виконуваний файл у той самий каталог, де знаходиться файл проєкту.

## 2.4.5. Запуск програми


Запуск програми може виконуватися з «*Головного*» меню **Run** (Запуск) обравши **Run** (Запуск) або натиснути на кнопку  **Run** (Запуск) панелі інструментів «*Debug*» (Налагодження) (рис.2.14) або натиснути клавішу **F9**.



Рис. 2.14. Панель інструментів «*Debug*» (Налагодження)

## 2.4.6. Помилки часу виконання

Під час роботи програми можуть виникати помилки, які називаються **помилками часу виконання** (run-time errors) або **винятками** (exceptions). У більшості випадків причинами **винятків** є невірні вихідні дані.

Після виникнення таких помилок програма перестане робити.

Необхідно перервати неробочу програму, вибравши із «*Головного*» меню **Run** (Запуск) команду **Program Reset** (Скидання програми) (рис. 2.15).

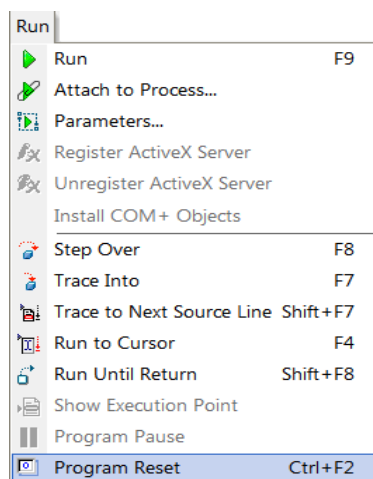


Рис. 2.15. Вибір команди **Program Reset** (Скидання програми) із «*Головного*» меню **Run** (Запуск)

## 2.4.7. Захист полів вводу

Для виключення помилок, при вводі даних у поля вводу необхідно передбачити у вихідному коді програми захист полів вводу.

У лабораторній роботі №8 передбачено захист полів вибору (див. п. п. 2.4.3).

### 3. Зміст звіту з лабораторної роботи №8

**Звіт** з лабораторної роботи №8 складається з:

- титульного аркуша (див. Додаток 3);
- аркуша (див. Додаток 3) на якому розташовано: **тема** лабораторної роботи, **мета** лабораторної роботи, **розробка програми** (наведіть умову, яка розглядається в лабораторній роботі №8, див. п. п. 2.4);
- аркуша (див. Додаток 3) на якому розташовано: **створення інтерфейсу програми** (навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №8, див. п. п. 2.4.1);
- аркушів (див. Додаток 3) на яких розташовані: **лістинг програми** (навести **лістинг** програми, який розглядається в лабораторній роботі №8, див. п. п. 2.4.3) та **висновки**.

### Запитання з самоконтролю до 3 розділу

1. Які оператори порівняння бувають?
2. Які умовні оператори бувають?
3. Які умовні оператори бувають за конструкцією?
4. Коли використовується оператор if?
5. Коли використовується оператор case?
6. Що таке вкладені оператори?

## ЛІТЕРАТУРА

### Основна

1. Программирование и отладка в Delphi. Учебный курс. Митчелл К. Керман М. : Вильямс, 2004. 720 с.
2. Библия для программиста в среде Delphi. М. Фленов. vr\_online@cydsoft.com, 2002. 572 с.
3. Основы Delphi. Профессиональный подход / М. В. Сухарев. СПб. : Наука и техника, 2004. 603 с.
4. Программирование в Delphi /А. Я. Архангельский. М. : Бином, 2003. 1152 с.
5. Delphi 6 / В. Э. Гофман, А. Д. Хомоненко. СПб. : БХВ Петербург, 2002. 628 с.
6. Delphi 6. Программирование на Object Pascal / Н. Б. Культин. М. : Мир, 2000. 524 с.

### Додаткова

7. Delphi. Профессиональное программирование / Д. Осипов. СПб. : Символ-плюс, 2006. 1056 с.
8. Delphi. Быстрый старт / В. Э. Гофман, А. Д. Хомоненко. СПб. : БХВ Петербург, 2003. 289 с.
9. Delphi 7. Справочное пособие / А. Я. Архангельский. М. : Бином, 2003.
10. Приемы программирования в Delphi / А.Я. Архангельский. М. : Бином, 2003.
11. Object Pascal в Delphi / А. Я. Архангельский. М. : Бином, 2002.

### Інформаційні ресурси в Інтернеті

12. Уроки для новичков – The Delphi. URL: <http://thedelphi.ru/lessons.php> (дата звернення: 10.02.2022 р.).
13. Уроки Delphi начинающим с нуля. URL: <http://www.delphi-manual.ru> (дата звернення: 10.02.2022 р.).
14. Ресурсы по Delphi. URL: <http://wiki.vspu.ru/users/bolgarochka22/delphi/index> (дата звернення: 10.02.2022 р.).
15. Delphi – Embarcadero Website. URL: <http://www.embarcadero.com> (дата звернення: 10.02.2022 р.).

**Додаток А**  
**Форма звіту з лабораторної роботи №1**

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ НАУКИ І ТЕХНОЛОГІЙ

Кафедра «Технологія машинобудування»

ЗВІТ

з лабораторної роботи №1  
«Блок-схема алгоритму з лінійною структурою»  
з дисципліни «Об'єктно-орієнтоване програмування»

Виконав

ст. гр. \_\_\_\_\_  
(№ групи)

\_\_\_\_\_  
(П.І.Б.)

Перевірив

\_\_\_\_\_  
(П.І.Б.)

Дніпро 20 \_\_

**Тема:**

**Мета роботи:**

**Складання блок-схеми алгоритму з лінійною структурою**

(навести умову та формулу, які розглядаються в лабораторній роботі №1, див. п. п. 2.3)

**Блок-схема алгоритму з лінійною структурою**

(навести блок-схему, яка розглядається в лабораторній роботі №1, див. п. п. 2.3)

**Висновки:**

**Додаток Б**  
**Форма звіту з лабораторної роботи №2**

**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ**  
**УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ НАУКИ І ТЕХНОЛОГІЙ**

**Кафедра «Технологія машинобудування»**

**ЗВІТ**

з лабораторної роботи №2  
**«Блок-схема алгоритму з розгалужуючою структурою»**  
з дисципліни «Об'єктно-орієнтоване програмування»

Виконав

ст. гр. \_\_\_\_\_  
(№ групи)

\_\_\_\_\_  
(П.І.Б.)

Перевірив

\_\_\_\_\_  
(П.І.Б.)

Дніпро 20 \_\_

**Тема:**

**Мета роботи:**

**Складання блок-схеми алгоритму з розгалужуючою структурою**  
(навести умову та формулу, які розглядаються в лабораторній роботі №2, див.  
п. п. 2.3)

**Блок-схема алгоритму з розгалужуючою структурою**  
(навести блок-схему, яка розглядається в лабораторній роботі №2, див.  
п. п. 2.3)

**Висновки:**

**Додаток В**  
**Форма звіту з лабораторної роботи №3**

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ НАУКИ І ТЕХНОЛОГІЙ

Кафедра «Технологія машинобудування»

ЗВІТ

з лабораторної роботи №3  
**«Лінійна структура програми та типи даних»**  
з дисципліни «Об'єктно-орієнтоване програмування»

Виконав

ст. гр. \_\_\_\_\_  
(№ групи)

\_\_\_\_\_  
(П.І.Б.)

Перевірив

\_\_\_\_\_  
(П.І.Б.)

Дніпро 20 \_\_\_\_

**Тема:**

**Мета роботи:**

**Розробка програми**

(навести умову та **формулу**, які розглядаються в лабораторній роботі №3, див. п. п. 2.4)

**Створення інтерфейсу програми та виведення повідомлень програми**

(навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №3, див. п. п. 2.4.1)

**Лістинг програми**

(навести **лістинг** програми, який розглядається в лабораторній роботі №3, див. п. п. 2.4.3)

**Висновки:**

**Додаток Г**  
**Форма звіту з лабораторної роботи №4**

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ НАУКИ І ТЕХНОЛОГІЙ

Кафедра «Технологія машинобудування»

**ЗВІТ**

з лабораторної роботи №4  
**«Розгалужуюча структура програми та типи даних»**  
з дисципліни «Об'єктно-орієнтоване програмування»

Виконав

ст. гр. \_\_\_\_\_  
(№ групи)

\_\_\_\_\_  
(П.І.Б.)

Перевірив

\_\_\_\_\_  
(П.І.Б.)

Дніпро 20 \_\_\_\_

**Тема:**

**Мета роботи:**

**Розробка програми**

(навести умову та формулу, які розглядаються в лабораторній роботі №4, див. п. п. 2.4)

**Створення інтерфейсу програми та виведення повідомлень програми**

(навести інтерфейс програми та вивід повідомлень програми, які розглядаються в лабораторній роботі №4, див. п. п. 2.4.1)

**Лістинг програми**

(навести лістинг програми, який розглядається в лабораторній роботі №4, див. п. п. 2.4.3)

**Висновки:**

**Додаток Д**  
**Форма звіту з лабораторної роботи №5**

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ НАУКИ І ТЕХНОЛОГІЙ

Кафедра «Технологія машинобудування»

ЗВІТ

з лабораторної роботи №5  
«Математичні функції та процедури»  
з дисципліни «Об'єктно-орієнтоване програмування»

Виконав

ст. гр. \_\_\_\_\_  
(№ групи)

Перевірив

\_\_\_\_\_  
(П.І.Б.)

\_\_\_\_\_  
(П.І.Б.)

Дніпро 20 \_\_\_\_

**Тема:**

**Мета роботи:**

**Розробка програми**

(навести умову та формулу, які розглядаються в лабораторній роботі №5, див. п. п. 2.4)

**Створення інтерфейсу програми та виведення повідомлень програми**

(навести інтерфейс програми та вивід повідомлень програми, які розглядаються в лабораторній роботі №5, див. п. п. 2.4.1)

**Лістинг програми**

(навести лістинг програми, який розглядається в лабораторній роботі №5, див. п. п. 2.4.3)

**Висновки:**

**Додаток Є**  
**Форма звіту з лабораторної роботи №6**

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ НАУКИ І ТЕХНОЛОГІЙ

Кафедра «Технологія машинобудування»

**ЗВІТ**

з лабораторної роботи №6  
**«Діалоговий ввід-вивід даних»**  
з дисципліни «Об'єктно-орієнтоване програмування»

Виконав

ст. гр. \_\_\_\_\_  
(№ групи)

\_\_\_\_\_  
(П.І.Б.)

Перевірив

\_\_\_\_\_  
(П.І.Б.)

Дніпро 20 \_\_\_\_

**Тема:**

**Мета роботи:**

**Розробка програми**

(навести **умову** та **формулу**, які розглядаються в лабораторній роботі №6, див. п. п. 2.4)

**Створення інтерфейсу програми та виведення повідомлень програми**

(навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №6, див. п. п. 2.4.1)

**Лістинг програми**

(навести **лістинг** програми, який розглядається в лабораторній роботі №6, див. п. п. 2.4.3)

**Висновки:**

**Додаток Ж**  
**Форма звіту з лабораторної роботи №7**

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ НАУКИ І ТЕХНОЛОГІЙ

Кафедра «Технологія машинобудування»

ЗВІТ

з лабораторної роботи №7  
**«Структури прийняття рішень. Оператор if»**  
з дисципліни «Об'єктно-орієнтоване програмування»

Виконав

ст. гр. \_\_\_\_\_  
(№ групи)

\_\_\_\_\_  
(П.І.Б.)

Перевірив

\_\_\_\_\_  
(П.І.Б.)

Дніпро 20 \_\_\_\_

**Тема:**

**Мета роботи:**

**Розробка програми**

(навести умову та формулу, які розглядаються в лабораторній роботі №7, див. п. п. 2.4)

**Створення інтерфейсу програми та виведення повідомлень програми**

(навести інтерфейс програми та вивід повідомлень програми, які розглядаються в лабораторній роботі №7, див. п. п. 2.4.1)

**Лістинг програми**

(навести лістинг програми, який розглядається в лабораторній роботі №7, див. п. п. 2.4.3)

**Висновки:**

**Додаток 3**  
**Форма звіту з лабораторної роботи №8**

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ НАУКИ І ТЕХНОЛОГІЙ

Кафедра «Технологія машинобудування»

ЗВІТ

з лабораторної роботи №8  
**«Структури прийняття рішень. Оператор case»**  
з дисципліни «Об'єктно-орієнтоване програмування»

Виконав

ст. гр. \_\_\_\_\_  
(№ групи)

Перевірив

\_\_\_\_\_  
(П.І.Б.)

\_\_\_\_\_  
(П.І.Б.)

Дніпро 20 \_\_\_\_

**Тема:**

**Мета роботи:**

**Розробка програми**

(навести умову, яка розглядається в лабораторній роботі №8, див. п. п. 2.4)

**Створення інтерфейсу програми та виведення повідомлень програми**

(навести інтерфейс програми та вивід повідомлень програми, які розглядаються в лабораторній роботі №8, див. п. п. 2.4.1)

**Лістинг програми**

(навести лістинг програми, який розглядається в лабораторній роботі №8, див. п. п. 2.4.3)

**Висновки:**